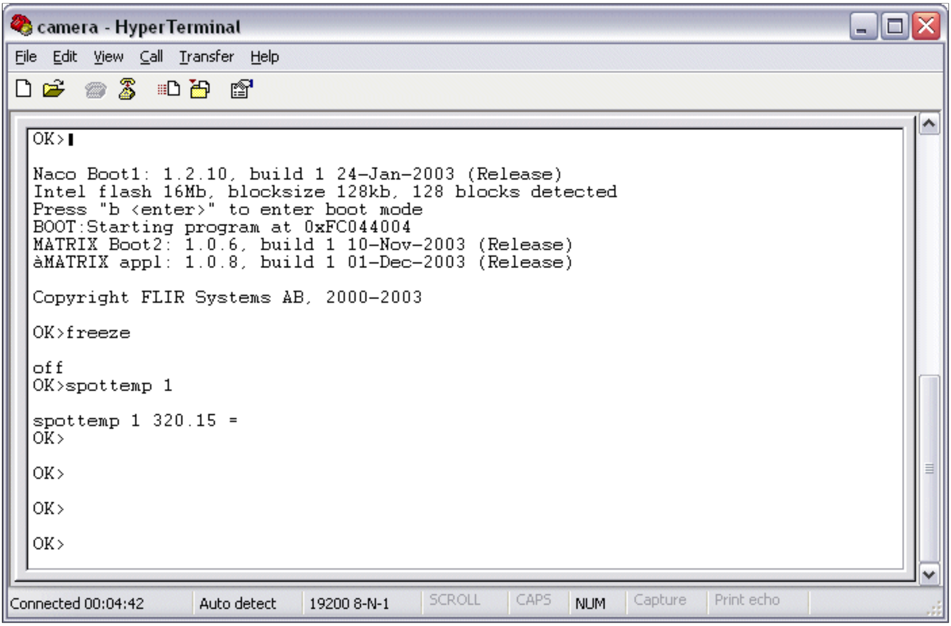
SXX & AXX commands

User’s Manual



```
OK>

Naco Boot1: 1.2.10, build 1 24-Jan-2003 (Release)
Intel flash 16Mb, blocksize 128kb, 128 blocks detected
Press "b <enter>" to enter boot mode
BOOT:Starting program at 0xFC044004
MATRIX Boot2: 1.0.6, build 1 10-Nov-2003 (Release)
àMATRIX appl: 1.0.8, build 1 01-Dec-2003 (Release)

Copyright FLIR Systems AB, 2000-2003

OK>freeze

off
OK>spottemp 1

spottemp 1 320.15 =
OK>

OK>

OK>

OK>
```

FLIR SYSTEMS™

Publ. No.	1 557 845
Revision	a40
Language	English (EN)
Issue date	March 3, 2004

SXX & AXX commands

User's Manual

FLIR SYSTEMS

CERTIFIED ISO 9001 SEMKO-DEKRA

Publ. No. 1 557 845 Rev. a40 – ENGLISH (EN) – March 3, 2004

Legal disclaimer

All products manufactured by FLIR Systems are warranted against defective materials and workmanship for a period of one (1) year from the delivery date of the original purchase, provided such products have been under normal storage, use and service, and in accordance with FLIR Systems instruction.

All products not manufactured by FLIR Systems included in systems delivered by FLIR Systems to the original purchaser carry the warranty, if any, of the particular supplier only and FLIR Systems has no responsibility whatsoever for such products.

The warranty extends only to the original purchaser and is not transferable. It is not applicable to any product which has been subjected to misuse, neglect, accident or abnormal conditions of operation. Expendable parts are excluded from the warranty.

In the case of a defect in a product covered by this warranty the product must not be further used in order to prevent additional damage. The purchaser shall promptly report any defect to FLIR Systems or this warranty will not apply.

FLIR Systems will, at its option, repair or replace any such defective product free of charge if, upon inspection, it proves to be defective in material or workmanship and provided that it is returned to FLIR Systems within the said one-year period.

FLIR Systems has no other obligation or liability for defects than those set forth above.

No other warranty is expressed or implied. FLIR Systems specifically disclaims the implied warranties of merchantability and fitness for a particular purpose.

FLIR Systems shall not be liable for any direct, indirect, special, incidental or consequential loss or damage, whether based on contract, tort or any other legal theory.

Copyright

Quality assurance

The Quality Management System under which these products are developed and manufactured has been certified in accordance with the ISO 9001 standard.

FLIR Systems is committed to a policy of continuous development; therefore we reserve the right to make changes and improvements on any of the products described in this manual without prior notice.

Patents

This product is protected by patents, design patents, patents pending, or design patents pending.

Postal address	FLIR Systems AB ■ P. O. Box 3 ■ SE-182 11 Danderyd ■ Sweden
Telephone	+46 (0)8 753 25 00
Telefax	+46 (0)8 753 23 64
Web site	www.flirthermography.com
E-mail	sales@flir.se

Table of contents

List of figures

1 About FLIR Systems

With over 30 years experience in IR systems and applications development, and over 30 000 infrared cameras in use worldwide, FLIR Systems is the undisputed global commercial IR industry leader.

10380703;2

Figure 1.1 FLIR Systems, Boston, USA, FLIR Systems, Danderyd, Sweden, and FLIR Systems, Portland, USA.

As pioneers in the IR industry, FLIR Systems has a long list of 'firsts' in the world of infrared thermography:

- 1965: 1st thermal imaging system for predictive maintenance (Model 650).
- 1973: 1st battery-operated portable IR scanner for industrial applications predictive maintenance (Model 750).
- 1975: 1st TV compatible system (Model 525).
- 1978: 1st dual-wavelength scanning system capable of real-time analog recording of thermal events (Model 780). Instrumental in R & D market development.
- 1983: 1st thermal imaging and measurement system with on-screen temperature measurement.
- 1986: 1st TE (thermo-electrically) cooled system.
- 1989: 1st single-piece infrared camera system for PM (predictive maintenance) and R & D (research & development) with on-board digital storage.
- 1991: 1st Windows-based thermographic analysis and reporting system.
- 1993: 1st Focal Plane Array (FPA) system for PM and R & D applications.
- 1995: 1st full-featured camcorder style FPA infrared system (ThermaCAM).
- 1997: 1st uncooled microbolometer-based PM/R & D system.
- 2000: 1st thermography system with both thermal and visual imaging.
- 2000: 1st thermography system to incorporate thermal/visual/voice and text data logging.
- 2002: 1st automated thermography system (model P60) to feature detachable remotely controllable LCD, JPEG image storage, enhanced connectivity including USB and IrDA wireless, thermal/visual/voice and text data logging.
- 2002: 1st low-cost ultra-compact hand-held thermography camera (E series). Revolutionary, ergonomic design, lightest IR measurement camera available.

- 2003: 1st low-cost, ultra-compact infrared camera for fixed installation for automation and security applications. Exceptionally user-friendly due to standard interfaces and extensive built-in functionality.

10401603;1

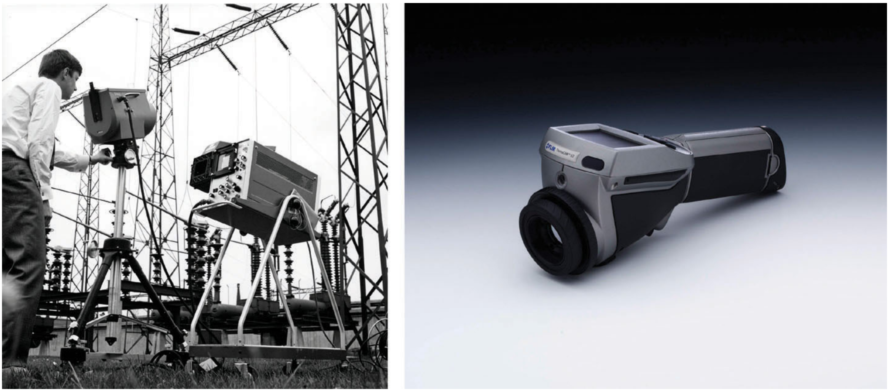


Figure 1.2 LEFT: FLIR Systems' Thermovision® Model 661. The photo is taken on May 30th, 1969 at the distribution plant near Beckomberga, in Stockholm, Sweden. The camera weighed approx. 25 kg (55 lb), the oscilloscope 20 kg (44 lb), the tripod 15 kg (33 lb). The operator also needed a 220 VAC generator set, and a 10 L (2.6 US gallon) jar with liquid nitrogen. To the left of the oscilloscope the Polaroid attachment (6 kg/13 lb) can be seen. **RIGHT:** FLIR Systems' ThermaCAM Model E2 from 2002 – weight: 0.7 kg (1.54 lb), including battery.

With this tradition of unparalleled technical excellence and innovative achievements, FLIR Systems continues to develop new infrared products, educational venues and applications expertise to meet the diverse demands of thermographers worldwide.

1.1 A few images from our facilities

10401303;1

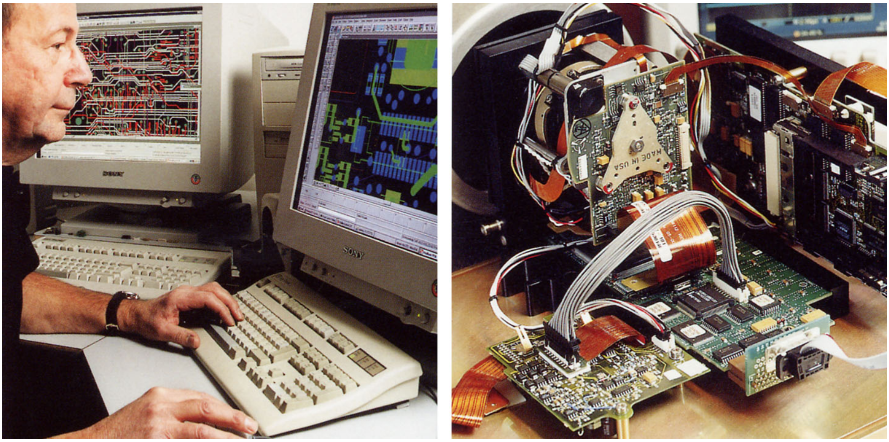


Figure 1.3 LEFT: Development of system electronics; **RIGHT:** Testing of an FPA detector

10401403;1

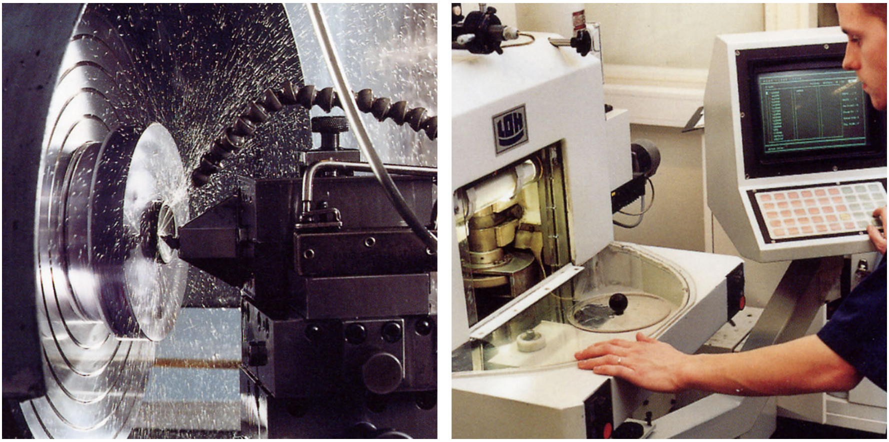


Figure 1.4 LEFT: Diamond turning machine; **RIGHT:** Lens polishing

10401503;1

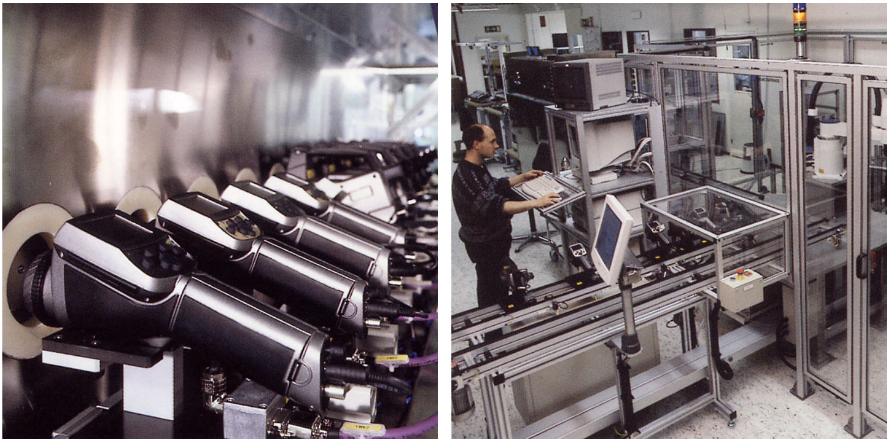


Figure 1.5 LEFT: Testing of IR cameras in the climatic chamber; **RIGHT:** Robot for camera testing and calibration

2 Typical system configurations

2.1 A20 & A40 system configurations

10406303;2

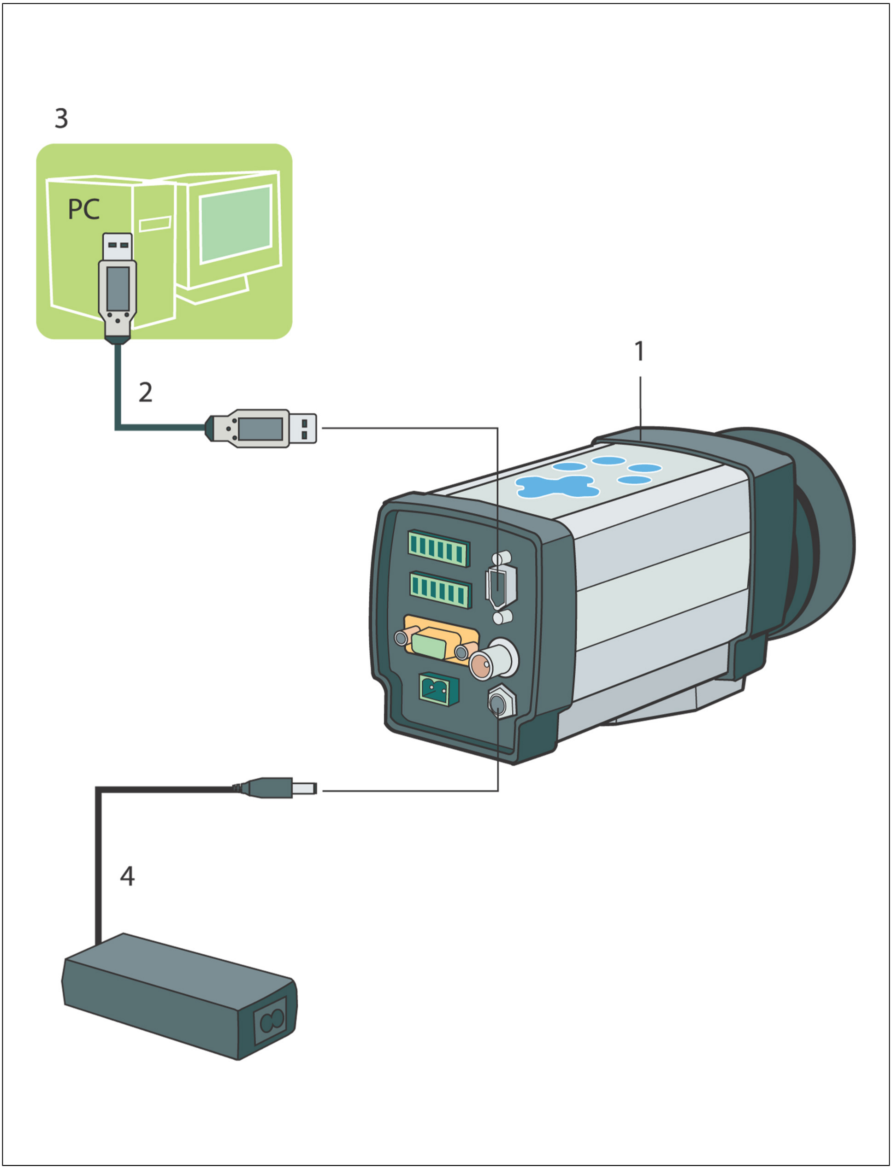


Figure 2.1 Typical system configurations: A20 V FireWire

10462803;1

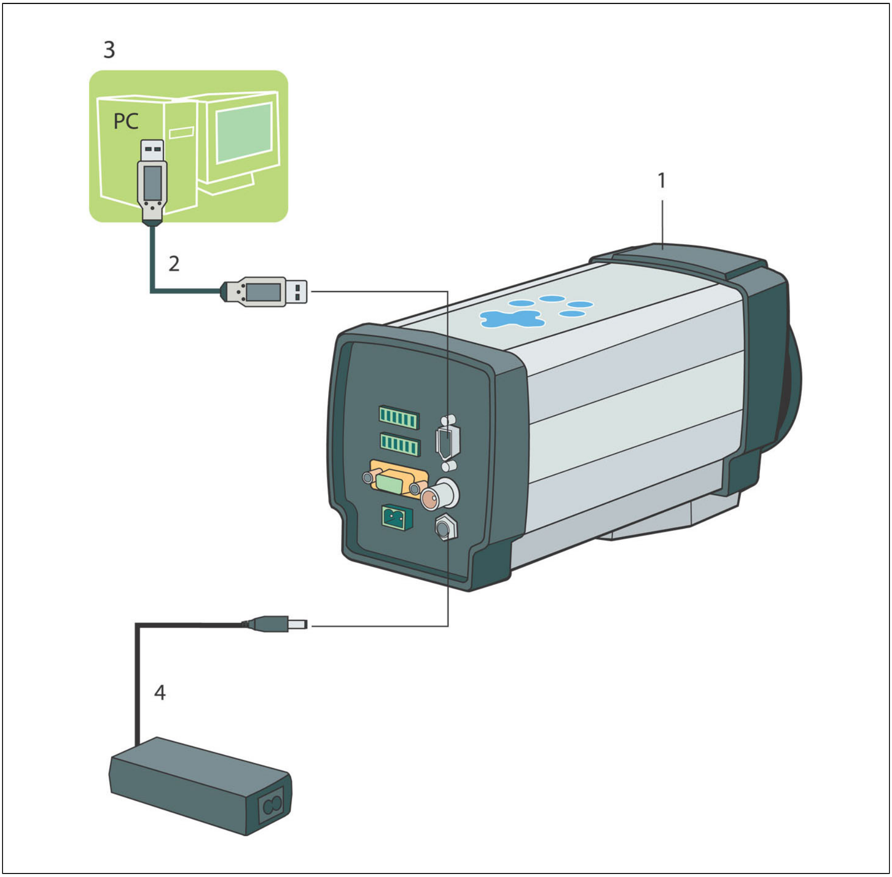


Figure 2.2 Typical system configurations: A40 V FireWire

Figure 2.3 Explanation of callouts

Callout	Explanation
1	Infrared camera
2	FireWire cable, 6-pin connector on camera FLIR P/N: ▪ 4/6-pin cable: 1 909 812 ▪ 6/6-pin cable: 1 909 921 ▪ 6/6-pin cable with screw locking in one end : 1 909 954
3	FLIR PC running: ▪ Application program developed with FLIR's SDK and Visual Basic/C++ (P/N 1 195 710)

Callout	**Explanation**
4	Power supply provided with the camera FLIR P/N 1 909 528 Camera only needs one power supply source

10406503;2

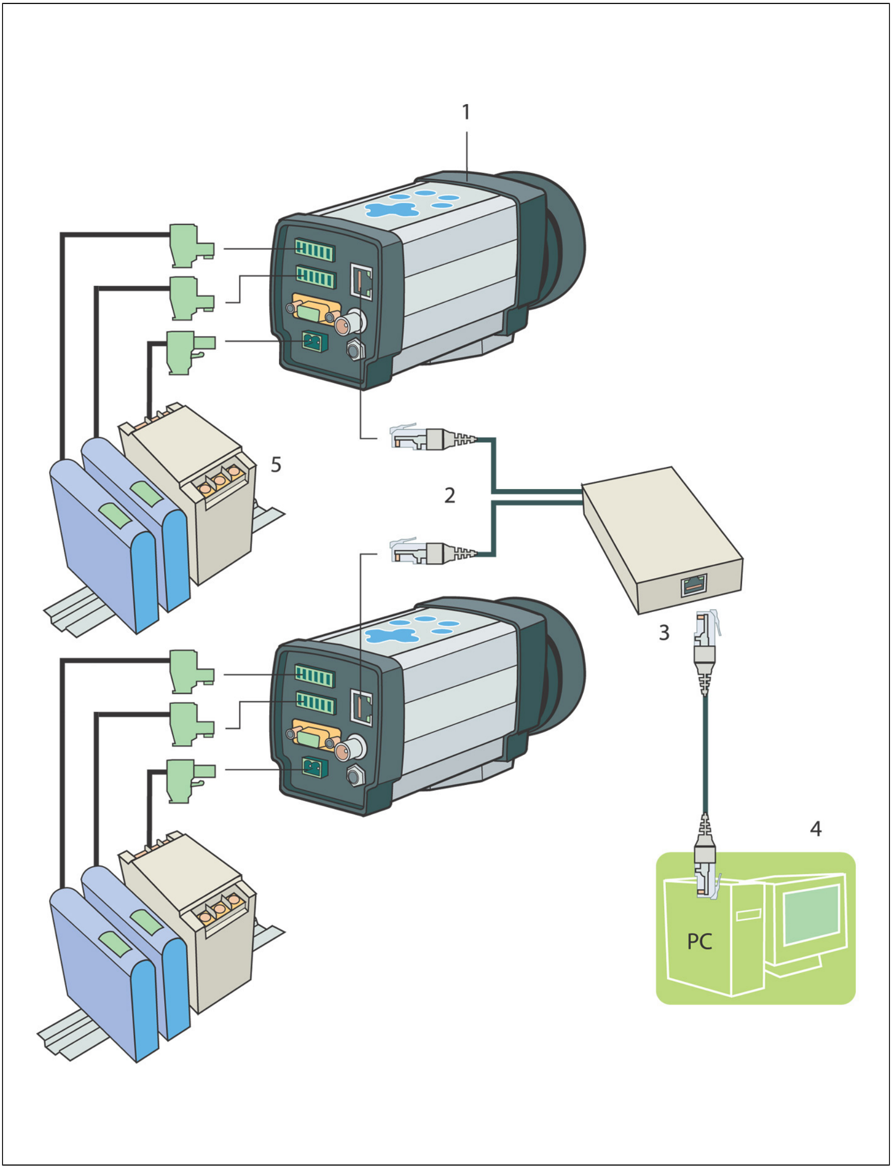


Figure 2.4 Typical system configurations: A20 V Ethernet

10463703;1

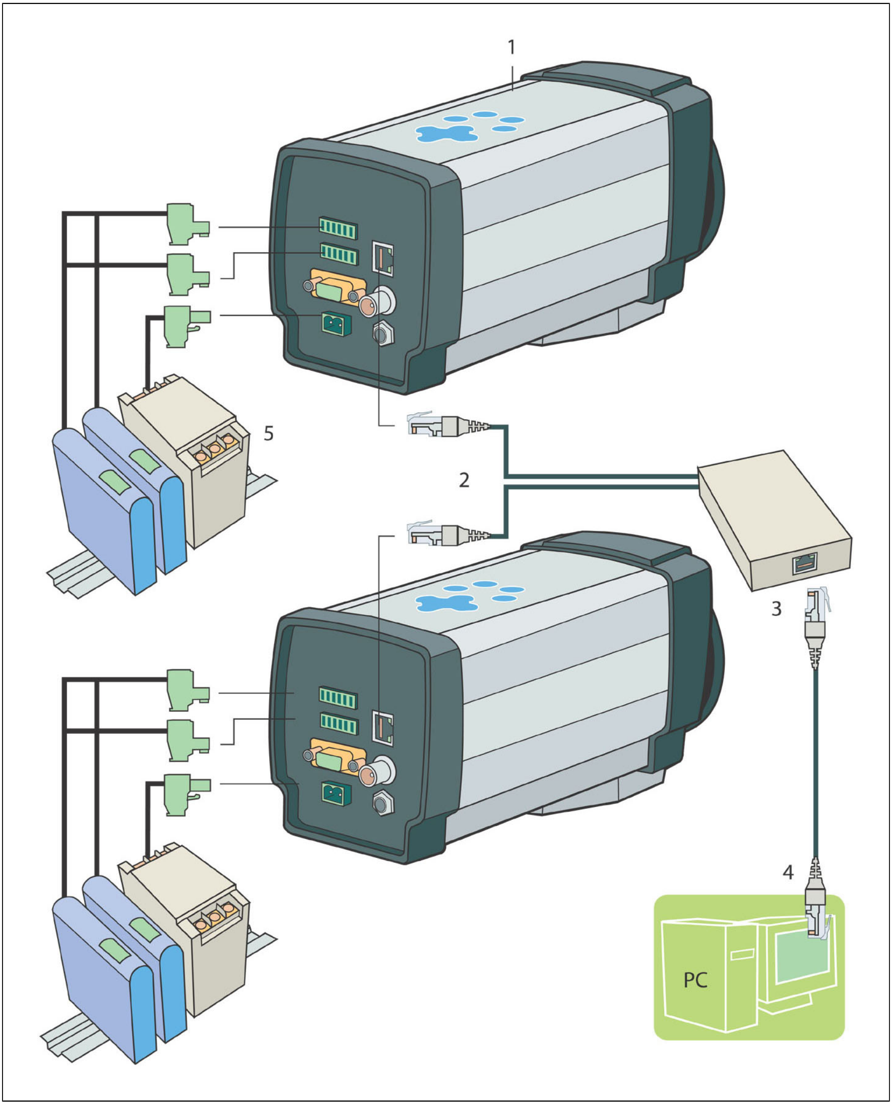


Figure 2.5 Typical system configurations: A40 V Ethernet

Figure 2.6 Explanation of callouts

Callout	Explanation
1	Infrared camera

Callout	Explanation
2	Ethernet cables RJ-45 FLIR P/N: ■ 2 m/6.56 ft straight with RJ-45 connector in both ends: 1 910 015 ■ 2 m/6.56 ft cross over with RJ-45 connector in both ends: 1 910 016
3	Ethernet switch
4	FLIR PC running: ■ Application program developed with FLIR's SDK and Visual Basic/C++ (P/N 1 195 710)
5	Industrial I/O-modules for isolation and power drive, preferably DIN rail mounted *Vendors:* Crouzet, Omron, Phoenix Contact, Weidmuller etc. **NOTE:** Power connector on camera is polarity protected.

10406303;2

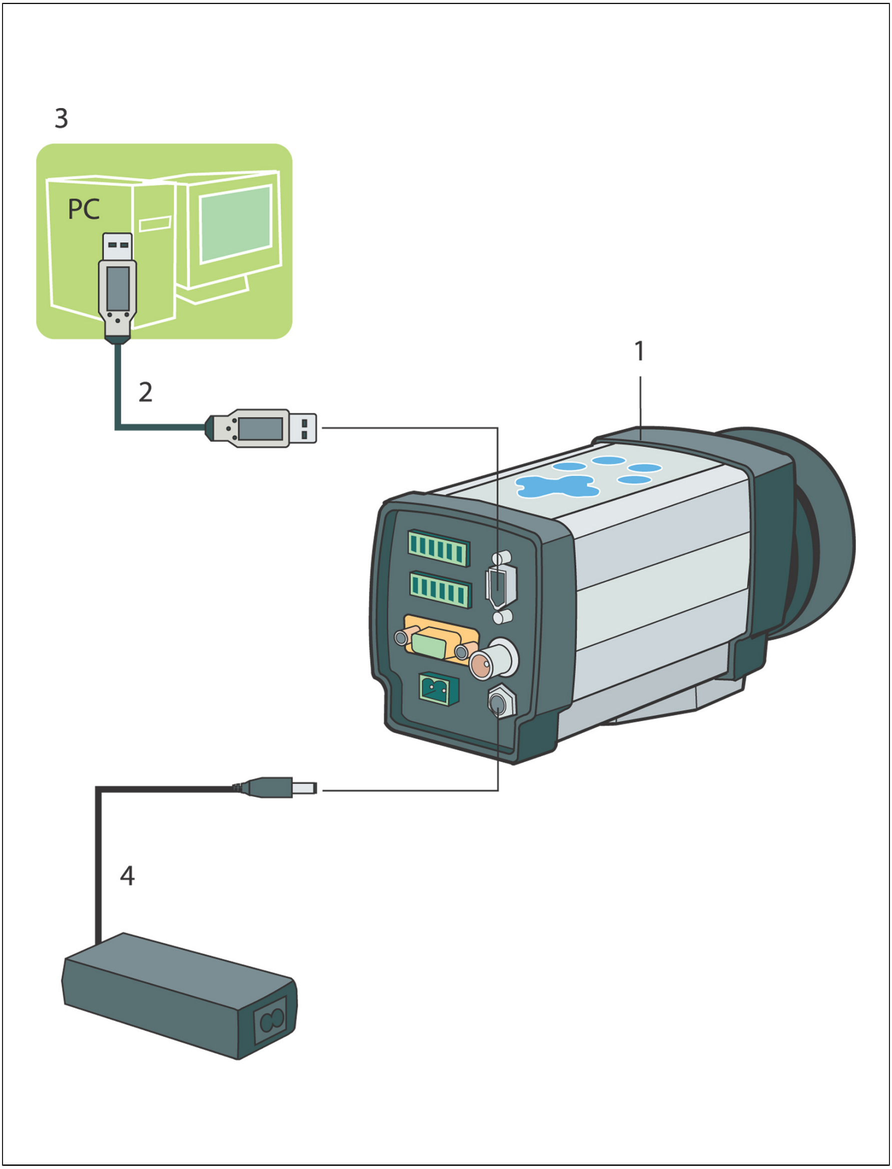


Figure 2.7 Typical system configurations: A20 M FireWire

10462803;1

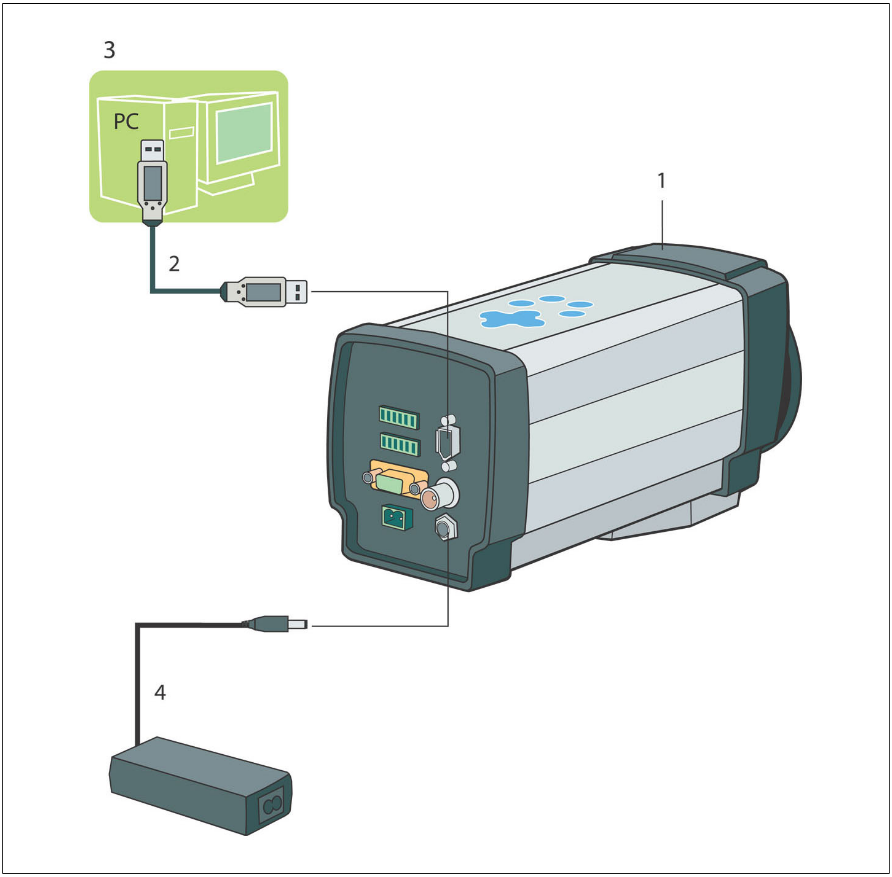


Figure 2.8 Typical system configurations: A40 M FireWire

Figure 2.9 Explanation of callouts

Callout	Explanation
1	Infrared camera
2	FireWire cable, 6-pin connector on camera FLIR P/N: ▪ 4/6-pin cable: 1 909 812 ▪ 6/6-pin cable: 1 909 921 ▪ 6/6-pin cable with screw locking in one end : 1 909 954
3	FLIR PC running: ▪ Application program developed with FLIR's SDK and Visual Basic/C++ (P/N 1 195 710)

Callout	Explanation
4	Power supply provided with the camera FLIR P/N 1 909 528 Camera only needs one power supply source

10406503;2

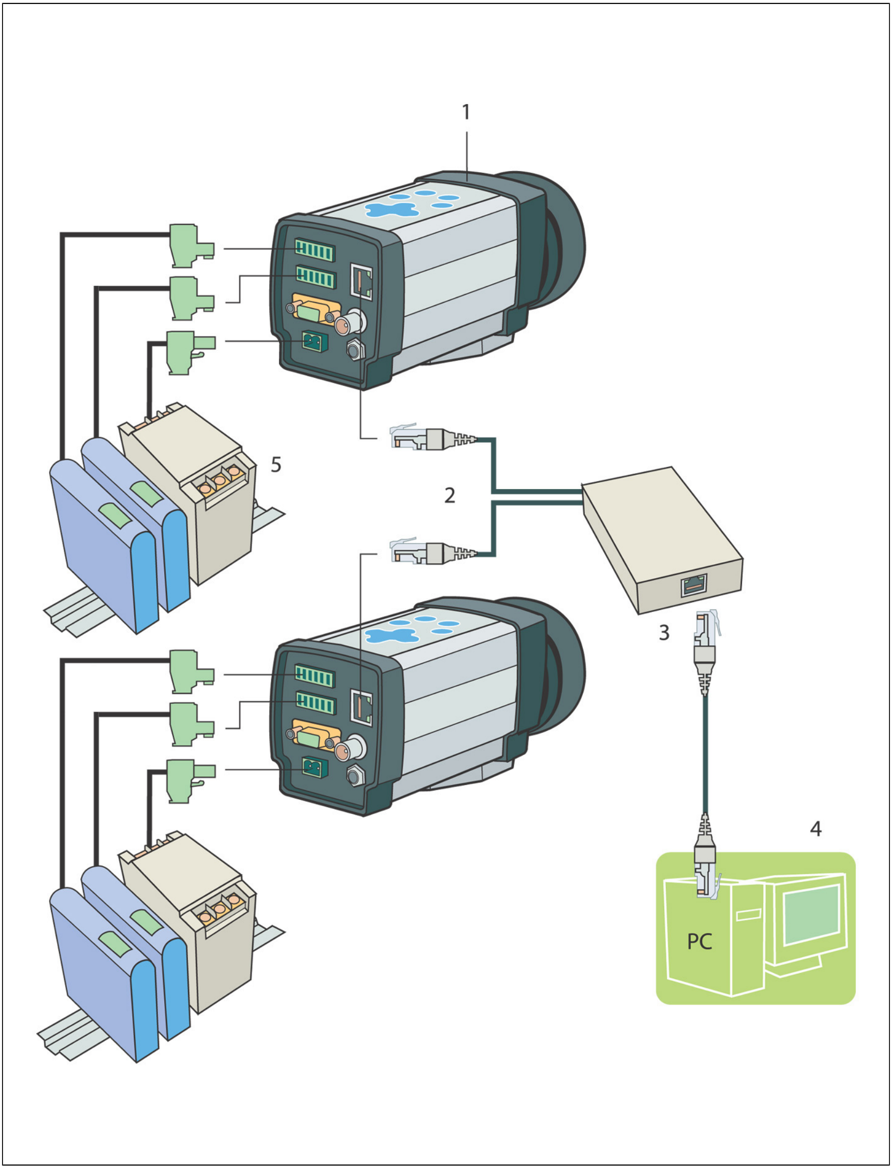


Figure 2.10 Typical system configurations: A20 M Ethernet

10463703;1

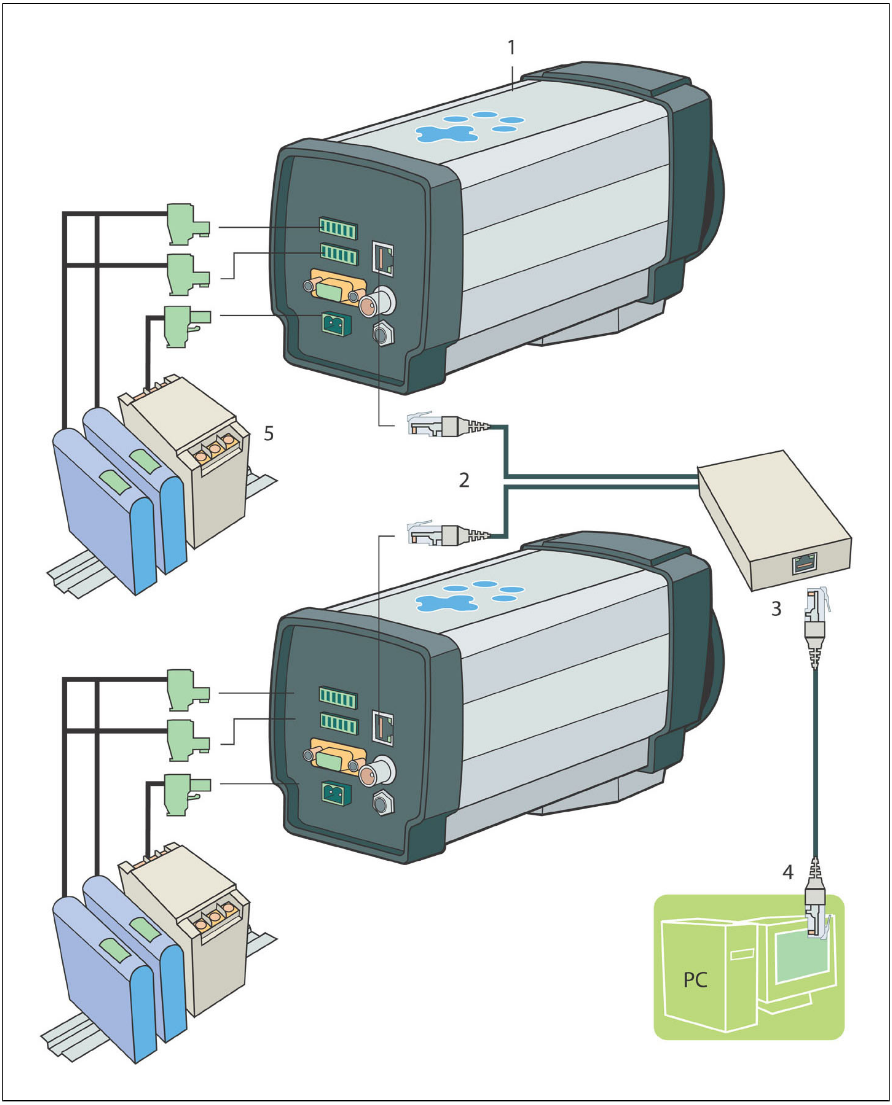


Figure 2.11 Typical system configurations: A40 M Ethernet

Figure 2.12 Explanation of callouts

Callout	Explanation
1	Infrared camera

Callout	Explanation
2	Ethernet cables RJ-45 FLIR P/N: ▪ 2 m/6.56 ft straight with RJ-45 connector in both ends: 1 910 015 ▪ 2 m/6.56 ft cross over with RJ-45 connector in both ends: 1 910 016
3	Ethernet switch
4	FLIR PC running: ▪ Application program developed with FLIR's SDK and Visual Basic/C++ (P/N 1 195 710)
5	Industrial I/O-modules for isolation and power drive., preferably DIN rail mounted *Vendors:* Crouzet, Omron, Phoenix Contact, Weidmuller etc. **NOTE:** Power connector on camera is polarity protected.

3 Connecting system components

10405903;2

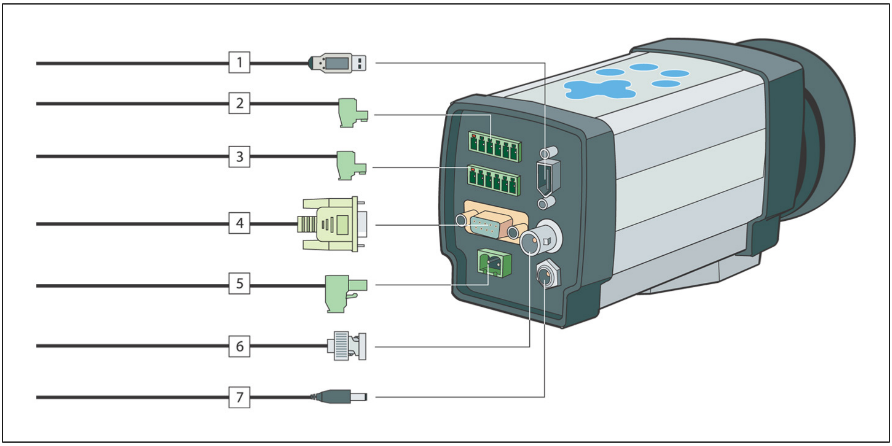


Figure 3.1 How to connect system components: A20 FireWire

10462903;1

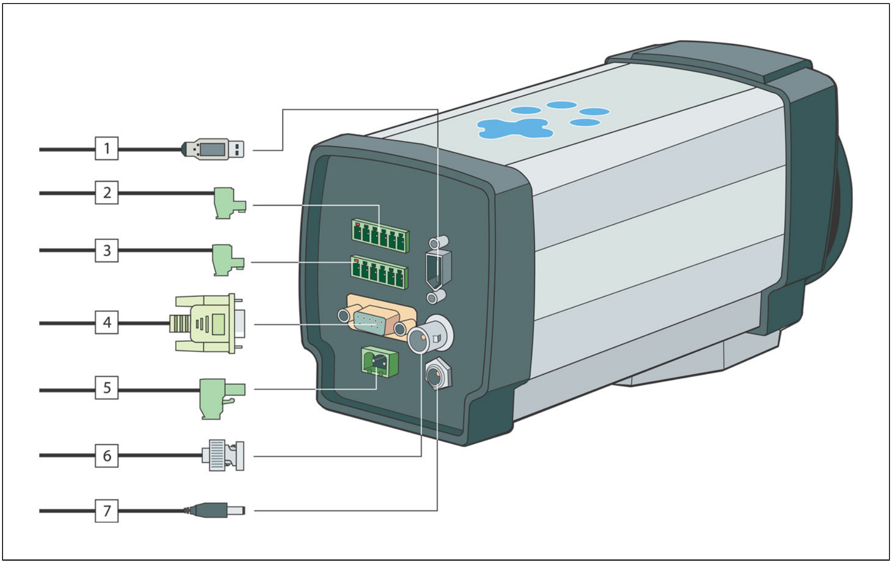


Figure 3.2 How to connect system components: A40 FireWire

Figure 3.3 Explanations of callouts

Callout	Explanation
1	FireWire cable. 6-pin connector on camera FLIR P/N: ▪ 4/6-pin cable: 1 909 812 ▪ 6/6-pin cable.:1 909 921 ▪ 6/6-pin cable with screw locking in one end: 1 909 954
2	Digital I/O-ports Jackable screw terminal *Vendor:* Phoenix Contact (www.phoenixcontact.com) P/N: 1803617 MC 1,5/6-ST-3,81
3	Analog I/O-ports Jackable screw terminal *Vendor:* Phoenix Contact (www.phoenixcontact.com) P/N: 1803617 MC 1,5/6-ST-3,81
4	RS-232 connection to computer. 9-pin, pin to pin cable
5	*A20 cameras:* Power supply. 12/24 V, minimum 15 W Recommended fuses (fast): 1 A (12 V); 500 mA (24 V) *A40 cameras:* Power supply. 12/24 V, minimum 20 W Recommended fuses (fast): 1.25 A (12 V); 630 mA (24 V) Jackable screw terminal *Vendor:* Phoenix Contact (www.phoenixcontact.com) P/N: 1757019 MSTB 2,5/2-ST-5,08 Camera only needs one power supply source **NOTE:** Power connector on camera is polarity protected.

Callout	Explanation
6	Video cable CVBS (composite video) out 75 ohm cable BNC to video monitor.
7	Power supply provided with the camera FLIR P/N: 1 909 528 Camera only needs one power supply source. **NOTE:** Power connector on camera is polarity protected.

10405803;2

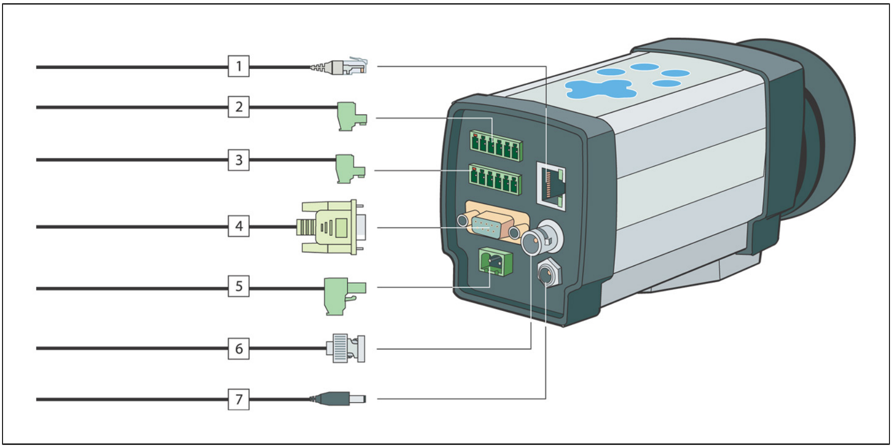


Figure 3.4 How to connect system components: A20 Ethernet

10463003;1

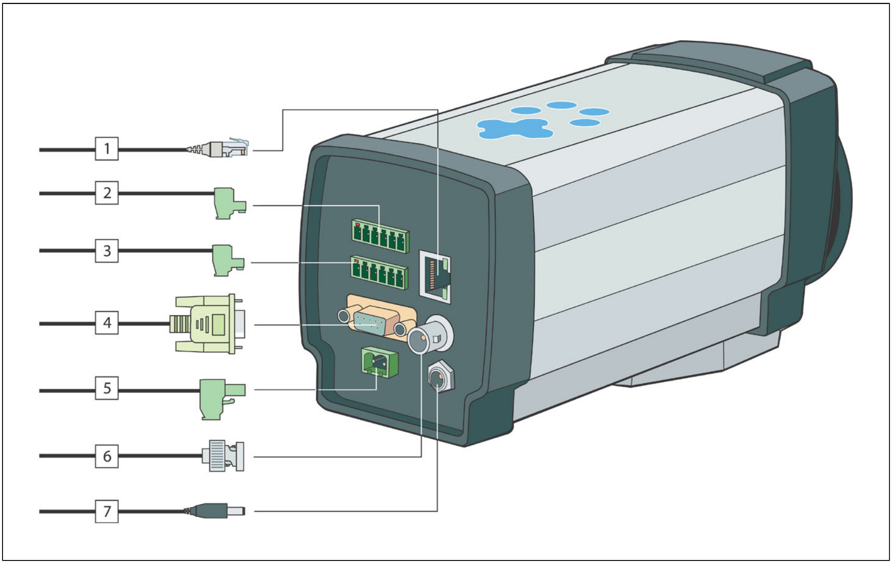


Figure 3.5 How to connect system components: A40 Ethernet

Figure 3.6 Explanations of callouts

Callout	Explanation
1	Ethernet cable RJ45
2	Digital I/O-ports Jackable screw terminal *Vendor:* Phoenix Contact (www.phoenixcontact.com) P/N: 1803617 MC 1,5/6-ST-3,81
3	Analog I/O-ports Jackable screw terminal *Vendor:* Phoenix Contact. (www.phoenixcontact.com) P/N: 1803617 MC 1,5/6-ST-3,81
4	RS-232 connection to computer. 9-pin, pin to pin cable

Callout	Explanation
5	*A20 cameras:* Power supply. 12/24 V, minimum 15 W Recommended fuses (fast): 1 A (12 V); 500 mA (24 V) *A40 cameras:* Power supply. 12/24 V, minimum 20 W Recommended fuses (fast): 1.25 A (12 V); 630 mA (24 V) Jackable screw terminal *Vendor:* Phoenix Contact (www.phoenixcontact.com) P/N: 1757019 MSTB 2,5/2-ST-5,08 Camera only needs one power supply source. **NOTE:** Power connector on camera is polarity protected.
6	Video cable CVBS (composite video) out 75 ohm cable BNC to video monitor.
7	Power supply provided with the camera FLIR P/N: 1 909 528 Camera only needs one power supply source. **NOTE:** Power connector on camera is polarity protected.

4 Connectivity overview

4.1 Physical interfaces

From the camera, there is a number of possible physical interfaces for data transfer.

- Serial communication
- FireWire (IEEE1394)
- Ethernet

Analog video, I/O ports and the keypad are not considered as data transfer interfaces

4.2 Low level protocols

On the mentioned physical interfaces, it is possible to run different low level protocols which varies slightly between the interfaces

4.2.1 RS-232

For RS-232 it is possible to run directly against a command interpreter (shell). Using this it is possible to type different commands from a console or a terminal emulation program, and get replies in text.

It is also possible to run tcp/ip over the serial port. Then a modem driver is used on a PC to establish a “PPP connection” in the same way that is done when connecting to internet using an ordinary modem (phone numbers not necessary). Performance is not that high (115200 baud). This is done when using for instance ThermaCAM Connect 2.0.

4.2.2 Ethernet

For Ethernet, only TCP/IP is supported. The camera should seamlessly work on any LAN, provided that a proper IP adress, netmask and possibly gateway is set in the camera.

No FLIR specific device drivers should be installed on the PC, also other types of computers and operating systems should work

The camera does not support dhcp naming for Ethernet, so a fixed IP address must be used.

4.2.3 FireWire

For FireWire, 2 types of traffic is possible in parallell.

- Isocronous image data using DCAM.

4 different framing formats are supported

- Asyncronous “FLIR 1394ip”

”TCP/IP over FireWire”

When connecting an A series camera over FireWire to a PC, the plug-and-play system will ask for 2 drivers. One for image data (DCAM) and one for network traffic (TCP/IP). Both of these drivers should be installed for full function.

In the PC, the camera will now look like 2 devices, one imaging device and one network adapter.

The camera contains a tiny dhcp service, so the camera may be either addressed by an automatic name (“ircamXXXXX” where XXXXX is the last 5 digits in the camera serial number) or directly using the corresponding ip number. It is also possible to manually configure the network adapter to use a specific ip number.

4.3 Functionality

4.3.1 Abstracted functionality

A lot of (but not all) software functionality is exposed through something called “software resources” .Those that are familiar with the Microsoft® Windows registry will recognize the idea.

However, in A-series (as well as E-/P-/S-Series) a resource node could also represent a software function that upon read or write actively interacts with the software system.

- Resources are organized in a tree hierarcy.
- A resource node could be only a data holder (for instance camera calibration)
- A resource node could be connected to a software function. (for instance temp sensor resources)
- A resource node has a type and certain attributes
- Many of the described services below interfaces with the system using the resource tree

4.3.2 Command control

Commands are read and interpreted by a “command shell”. One instance of the command shell runs against the RS-232 port after startup and interprets commands typed here. Some commands are “standard commands” like for instance “ls”, “cd” that runs directly against the system. Other commands relies upon resource access, for instance the “emissivity” command. There are also some direct resource commands, like “rls”, “rset” for direct access to the resource hierarcy.

It is possible to run additional, independent instances of the command shell when using the “telnet” service on an established tcp/ip connection. In a telnet session, almost all commands could be used in the same way as when typed into the RS232 shell, except for some special ones, like “baudrate”.

4.3.3 IP services

Independent of physical interface, it is possible to access the system using tcp/ip using the exposed services described here: telnet, ftp, http, CIFS, FLIR resource socket, FLIR RTP. More than one service and possibly more than one instance of the service may be run at the same time “simultaneously”, for instance 2 telnet’s and one WS_FTP together against the same camera.

4.3.3.1 telnet

Command control, mainly for manual typing. Typical clients are the “standard” telnet command on a PC or “tera term”.

4.3.3.2 ftp

File transfer to/from the camera using an ftp client software on a PC. Typical clients are the “standard” ftp command on a PC, “WS_FTP” and also “ThermaCAM Connect 2.0”

4.3.3.3 http

Web server. Typical clients are Microsoft internet explorer and Netscape.

4.3.3.4 CIFS

PC network file access service. This service makes it possible to map a drive letter on the PC to the camera file system. The intended client software is built into all relevant windows versions.

4.3.3.5 FLIR resource socket

It is possible to read and write nodes of the software resource tree from a PC client software directly. Then the PC client software should read and write resources using a standard socket. There are no “standard” clients available.

4.3.3.6 FLIR RTP

RTP stands for “real time protocol”. It is a “standard” for sending streaming data over a socket. The A-Series uses RTP for the “Image viewer” functionality of the user web site using a java client. This client code is exported from the camera itself, but it requires some supporting Java components installed in the PC. FLIR RTP is also used within the ThermoVision LabVIEW® Toolkit / ThermoVision SDK.

For the moment the RTP protocol is for FLIR propriety usage only (not public).

4.3.3.7 DNS, dhcp, netBIOS

There are “tiny” implementations of these services. DNS is only active on 1394ip (and USB).

5 Communication Protocol Specification

5.1 Introduction

This section describes the command interface used on the serial communication port or by telnet in the camera application.

5.2 Low level protocol

RS-232 is used as physical interface and low-level protocol. The camera board acts as a Data Communication Equipment (DCE). The opposite communication part (the PC or controlling computer) act as a Data Terminal Equipment (DTE) (and normally has a male 9 pin D-SUB connector).

Default communication parameters are:

Baud rate:	19200
Data bits:	8
Parity:	None
Stop bits:	1
Flow control:	None

5.3 Command syntax description

The basic idea is that the command protocol shall be easy to interpret by both human users and machines. The commands and any response that are sent over the communication channel, only uses ASCII printable characters. The user interface is not so different from several well-known standards used for terminal sessions (e.g. UNIX-shell command interpreter). In fact, this protocol is implemented with the help of pSOS shell commands.

5.3.1 Commands

The commands (or system input) follow the syntax:

```
<command> <WS> <parameters> <CR>
```

where:

`<command>`	is the character string identifying one of the commands described in detail below,

`<parameters>`	specifies one or more parameters to the command, all separated by <WS>, and possibly identified with special flag parameters e.g. –y.
`<WS>`	white space - command and parameters are separated with one or many ASCII Space or Tab characters,
`<CR>`	specifies the one-byte ASCII control character CR that ends each command request.

The command can also be a query for the current board setting:

```
<command> <CR>
```

5.3.1.1 Parameters

Valid parameters can be of the following types:

Integer	ASCII string representation of an integer value. Space, dot or comma is not allowed (E.g. 115200).
String	ASCII string (E.g. off). If the string contains more than one word, the string has to bee delimited by double quotation marks. E.g. "Like this!"
Character	ASCII character (E.g. +).
Flag	two characters combination starting with a '-' (E.g. –y), identifying the following parameter(s).
Float	ASCII string representation of a decimal value. Dot is used to separate decimals (E.g. 0.5)

5.3.1.2 Example

Examples of how normal commands would look like:

```
level 1000 <CR>

rm -r web/service <CR>
```

5.3.2 Responses

If the command has parameters to alter the current camera settings the response (system output) is:

```
<CR><LF>OK>
```

where:

`<CR><LF>`	specifies the two-byte ASCII control characters CR and LF that ends every line of output data

OK>	is the shell prompt indicating that the board is ready to receive next command.

If the command is a query, typically a command without any parameters, there will be a response with the following syntax:

```
<CR><LF><result><CR><LF>OK>
```

where:

<result>	is one or many results given by that command, all separated by <WS>, and if the result takes more than one line there will be a <CR><LF> line break inserted between every line.

The camera board always responds to a command with a <CR><LF> and the “OK>” prompt to announce that it is ready to run next command.

This is also the case for an empty command (a single <CR>). This can be used to check if camera board is ready and running. Note: Incoming commands are buffered and input buffers can overflow if commands are sent too frequently. (1 Hz is not a problem.)

5.3.2.1 Example

The command:

```
level<CR>
```

could give the following result:

```
<CR><LF>1000<CR><LF>OK>
```

while the command:

```
level 1000<CR>
```

will get the shorter response:

```
<CR><LF>OK>
```

5.3.2.2 Initial response

The camera board initially responds to power up with an initial message followed by the “OK>” prompt:

```
<Initial Message String><CR><LF>OK>
```

where the <Initial Message String> is an undefined character string (not containing <CR><LF>OK>).

When the <CR><LF>OK> prompt is received the board is ready to receive the first command.

5.3.3 Error responses

If the command is somehow erroneous received there is a response indicating the error. Depending on the kind of error the responses will be:

`<CR><LF><errcmd>:<Sp>Command not found`	when a command not is recognized by the interpreter,
`<CR><LF><errcmd>:<Sp>Command ambiguous`	when the interpreter don't get an unique match to the command given, (It is possible to activate a command without typing all of the characters as long as there isn't another command which starts with the same string.)
`<CR><LF><errcmd>:<Sp>Error:<Sp><ErrNo>, <Sp><reason>`	when a parameter to the command is of wrong type or out of range, or if the command failed by other reasons, e.g. hardware not ready.

Above <errcmd> is an erroneous command, `<Sp>` is the ASCII "Space" character, `<ErrNo>` is the integer identifying the error and <reason> is an optional text string that explains the error.

When the `<CR><LF>OK>` prompt is received again the board is ready for the next command.

5.4 Command definitions

The following command definitions follows the syntax described above under , but for reasons of readability, when specified in this section, it will not be at the same level of detail.

The “OK>” prompt, the “CR” the “LF”, the “space” and the “white space” will all be there even if omitted in the following sections.

Parameters enclosed in brackets are optional, e.g. [-o <option>] where –o is an optional flag and <option> is its parameter.

Response is only specified for the case of a query command.

5.4.1 pSOS pSH+ specified commands

The following commands are specified by pSOS. Most of them works as UNIX commands, but often with reduced possibilities to use flags.

SEE ALSO: For a detailed description of these commands, see section 5.5 – pSH+ Built-In Commands on page 54.

The detailed description of these commands can be found in the pSOS manual “Concepts and User’s Guide”, chapter 11.9.6 “pSH+ Built in commands”..

`arp`	Address resolution display and control.
`cat`	Concatenate and display files.
`cd`	Change working directory.
`clrscr`	Clear terminal window.
`cmp`	Compare two files.
`cp`	Copy files.
`date`	Display or set date.
`du`	Display disk block usage.
`echo`	Echo arguments to the standard output.
`getpri`	Get task priority.
`head`	Display the first few lines of a file.
`help`	Display reference manual pages.
`history`	Display a previous command, using the command number.
`ifconfig`	Configure network interface parameters.

`kill`	Terminate a task.
`ls`	List the contents of a directory.
`mkdir`	Create a directory.
`mount`	Mount a pHILE+ file system.
`mv`	Move or rename files.
`netstat`	Show network status.
`ping`	Send ICMP ECHO REQUEST packets to network host.
`popd`	Pop the directory stack.
`pushd`	Push current directory onto the directory stack.
`pwd`	Display pathname of the current working directory.
`resume`	Resume a task.
`rm`	Remove files.
`rmdir`	Remove directories.
`route`	Manually manipulate the routing tables.
`setenv`	Set environment variables.
`setpri`	Set task priority.
`sleep`	Suspend execution for a specified interval.
`suspend`	Suspend a task.
`sync`	Force all changed blocks to disk.
`tail`	Display the last part of a file.
`touch`	Update the modification time of a file.
`umount`	Unmount a file system.

5.4.2 System commands

5.4.2.1 Restart

This command resets the system and makes a complete restart. This is useful if a new application is loaded.

Syntax	`restart`
Param.	No parameters
Query res.	No response (not even OK, as the system resets).

5.4.2.2 Baud rate

Set/display the baud rate on the serial interface. Changing the baud rate has a temporary effect (until next reboot).

<table>
<tr><td>Syntax</td><td><code>baudrate [-p <portno>] <rate&gt;</code>
<code>baudrate [-p <portno&gt;]</code></td></tr>
<tr><td>Param.</td><td>
<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td><code>&lt;portno&gt;</code></td><td>Integer</td><td>1 (default) or 2</td></tr>
<tr><td><code>&lt;rate&gt;</code></td><td>Integer</td><td>9600, 19200, 38400, 57600, 115200</td></tr>
</table>
</td></tr>
<tr><td>Query res.</td><td><code>&lt;rate&gt;</code></td></tr>
</table>

5.4.2.3 Terminal echo

Sets the status of the terminal echo, i.e. if characters typed should be echoed back. By default terminal echo is on.

<table>
<tr><td>Syntax</td><td><code>termecho [-p <portno>] <state&gt;</code>
<code>termecho [-p <portno&gt;]</code></td></tr>
<tr><td>Param.</td><td>
<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td><code>&lt;portno&gt;</code></td><td>Integer</td><td>1 (default) or 2</td></tr>
<tr><td><code>&lt;state&gt;</code></td><td>String</td><td>on, off (default: on)</td></tr>
</table>
</td></tr>
<tr><td>Query res.</td><td><code>&lt;state&gt;</code></td></tr>
</table>

5.4.2.4 PPP

Initiate PPP (TCP/IP) communication over RS-232 interface

<table>
<tr><td>Syntax</td><td><table>
<tr><td>ppp &lt;rate&gt;</td><td>Restore old baudrate after completion. Simulate modem.</td></tr>
<tr><td>ppp</td><td></td></tr>
<tr><td>flirppp &lt;rate&gt;</td><td>Set 19200 baud after completion. Do not simulate modem.</td></tr>
<tr><td>flirppp</td><td></td></tr>
</table></td></tr>
<tr><td>Param.</td><td><table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td>&lt;rate&gt;</td><td>Integer</td><td>9600, 19200, 38400, 57600, 115200</td></tr>
</table></td></tr>
<tr><td>Query res.</td><td>CONNECT &lt;rate&gt;</td></tr>
</table>

5.4.2.5 PPP address

Set IP address to be distributed by camera at next PPP connection.

<table>
<tr><td>Syntax</td><td>pppaddr <cameraaddr> <pcaddr>
pppaddr</td></tr>
<tr><td>Param.</td><td><table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td>&lt;cameraaddr&gt;</td><td>String</td><td>IP address of camera (e.g. 192.168.0.1)</td></tr>
<tr><td>&lt;pcaddr&gt;</td><td>String</td><td>IP address of host (e.g. 192.168.0.2)</td></tr>
</table></td></tr>
<tr><td>Query res.</td><td>Camera address <cameraaddr> and PC address <pcaddr&gt;</td></tr>
</table>

5.4.2.6 RTC date and time

Set real time clock of camera.

Syntax	rtcdate <yymmddhhmmss>

<table>
<tr><td>Param.</td><td><table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td>&lt;yymmddhhmmss&gt;</td><td>Packed integers</td><td>yy – Year 00-99
mm – Month 01-12
dd – Date 01-31
hh – Hour 00-23
mm – Minute 00-59
ss – Second 00-59</td></tr></table></td></tr>
<tr><td>Query res.</td><td>No query is defined.</td></tr>
<tr><td>Scope</td><td>rtcdate command is available in appl mode only</td></tr>
</table>

5.4.2.7 sysvars (A-Series only)

NOTE: This is a very sensitive command and should only be used by personnel with the appropriate knowledge.

This command gives possibility to display and change LAN (ethernet) parameters on a low level.

Syntax	`sysvars`
	`sysvars [-m <mode>][-t <type>][-d <delay>][-b <baudrate>][-e <LANenable-flag>]`
	`[-i <ipnumber>][-n <mask>][-g <gateway ip>]`
	`sysvars –default`

Param.

Parameter	**Type**	**Value**
`<mode>`	Integer	debug mode (do not touch)
`<type>`	Integer	remote debug type (do not touch)
`<delay>`	Integer	Debug timeout (do not touch)
`<baudrate>`	Integer	Possible default baud rate value. Tells startup baudrate (change with care)
`<LANenableflag>`	Integer	1 – LAN (ethernet) enabled 0 – LAN (ethernet) disabled
`<ipnumber>`	String	syntax “x.x.x.x” where x is a number 0-255. IP number to use on ethernet interface
`<mask>`	String	syntax “x.x.x.x”. IP mask to use on ethernet interface
`<gateway ip>`	String	syntax “x.x.x.x”. IP number for default gateway. “0.0.0.0” if not used
`“-default”`	String	Set all values to default

Query res.

Typical output from “sysvars”

```
OK>sysvars
Magic value : OK
Checksum : OK
Debug mode : 4
Remote debug type : 0
Startup delay : 3
Default baudrate : 19200
Lan enabled : 0
Lan ip address : 10.4.2.20
Lan netmask : 255.255.0.0
Default gateway : 0.0.0.0
OK>
```

5.4.3 Camera control commands

5.4.3.1 Non uniformity correction (NUC)

Initiates the calculation of a new offset map. The result of the NUC process is the replacement of one offset map in SDRAM with the newly generated map. The number of frames to average over is given by calculating 2 powers <exp>, (2^<exp>) where <exp> is the parameter given to the command. This allows the range of 1-128 frames to average over.

<table>
<tr><td>Syntax</td><td><table><tr><td><code>nuc [&lt;exp&gt;]</code></td><td>If no parameter given default is 5 which gives a 32 frames average.</td></tr><tr><td><code>nuc -n [&lt;exp&gt;]</code></td><td>Same as above but without activating the shutter.</td></tr></table></td></tr>
<tr><td>Param.</td><td><table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td><code>&lt;exp&gt;</code></td><td>Integer</td><td>0-7 (Default = 5)</td></tr></table></td></tr>
<tr><td>Query res.</td><td>No query is defined.</td></tr>
<tr><td>Scope</td><td>nuc command is available in appl mode only</td></tr>
</table>

5.4.3.2 Video mode

Changes or reads the current video mode.

<table>
<tr><td>Syntax</td><td><table><tr><td><code>videomode <mode&gt;</code></td><td>Set video mode</td></tr><tr><td><code>videomode</code></td><td>Read video mode.</td></tr></table></td></tr>
<tr><td>Param.</td><td><table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td><code>&lt;mode&gt;</code></td><td>String</td><td>NTSC or PAL</td></tr></table></td></tr>
<tr><td>Query res.</td><td><code>&lt;mode&gt;</code></td></tr>
<tr><td>Scope</td><td>videomode command is available in appl mode only</td></tr>
</table>

5.4.3.3 Automatic Focus Adjust

Controls the automatic focus capability of the camera.

Syntax	`autofocus <cmd>`
	`autofocus`

<table>
<tr><td>Param.</td><td>
<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td>&lt;cmd&gt;</td><td>String</td><td>now</td></tr>
<tr><td>&lt;state&gt;</td><td>String</td><td>off, on. ‘on’ if autofocus in progress, ‘off’ if it has finished.</td></tr>
</table>
</td></tr>
<tr><td>Query res.</td><td>&lt;state&gt;</td></tr>
<tr><td>Scope</td><td>autofocus command is available in:
▪ P-Series with appl version 2.4.6 or later
▪ A-Series (A40) with appl version 1.0.8 or later</td></tr>
</table>

5.4.3.4 Focus

Controls the camera focus. ThermoVision A20 with motorfocus does not provide absolute positions, and therefore only supports the –i, -c, -s arguments, and no queries.

<table>
<tr><td>Syntax</td><td>
<table>
<tr><td>focus –i<speed&gt;</td><td>Runs the focus motor towards infinity.</td></tr>
<tr><td>focus –c <speed&gt;</td><td>Runs the focus motor towards close-up distance.</td></tr>
<tr><td>focus –s</td><td>Stops the current focus move at current position</td></tr>
<tr><td>focus –r <steps&gt;</td><td>Relative focus move in steps (positive value towards infinity and negative value towards close-up).</td></tr>
<tr><td>focus –a &lt;abs&gt;</td><td>Moves focus to an absolute focus position.</td></tr>
<tr><td>focus –q</td><td>Query focus limits, will report lower and upper limits respectively.</td></tr>
<tr><td>focus</td><td>Query of absolute position.</td></tr>
</table>
</td></tr>
<tr><td>Param.</td><td>
<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td>&lt;speed&gt;</td><td>Integer</td><td>0-100 (Speed in percent of full speed)</td></tr>
<tr><td>&lt;steps&gt;</td><td>Integer</td><td>Depends on limits for actual hardware and
current position.</td></tr>
<tr><td>&lt;pos&gt;</td><td>Integer</td><td>Depends on limits for actual hardware. Greater value towards infinity.</td></tr>
</table>
</td></tr>
</table>

Query res.	Only for cameras that with focus that provide absolute focus positions.
	`focus:`
	`<pos>`
	`focus –q:`
	`<pos (low)> <pos (high)>`

Scope	Focus command is available in: ▪ P-Series with appl version 2.4.6 or later ▪ A-Series (A40 + A20 with motor focus option) with appl version 1.0.8 or later

5.4.3.5 Temperature Range

This command sets the working mode of the imager. It controls the temperature range.

<table>
<tr><td>Syntax</td><td><code>temprange <highTemp&gt;</code>
<code>temprange</code></td></tr>
<tr><td>Param.</td><td><table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td><code>&lt;highTemp&gt;</code></td><td>Integer</td><td>High temperature of desired range in Celsius.</td></tr></table></td></tr>
<tr><td>Query res.</td><td><code>&lt;state&gt;</code>
Gives a list of all selectable temperature ranges. The currently selected range is marked with an asterix (*).</td></tr>
<tr><td>Scope</td><td>Temprange command is available in:
▪ P-Series with appl version 2.4.6 or later
▪ A-Series with appl version 1.0.8 or later</td></tr>
</table>

5.4.3.6 Latchmode

This command sets or reads the working latchmode mode of the imager. Refers to **Image Settings** → **Lock To** in the ThermoVision A20 and A40 camera GUI.

<table>
<tr><td>Syntax</td><td><code>latchmode <mode&gt;</code>
<code>latchmode</code></td></tr>
<tr><td>Param.</td><td><table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td><code>&lt;latchmode&gt;</code></td><td>String</td><td>Pixel/power/temp</td></tr></table></td></tr>
<tr><td>Query res.</td><td><code>&lt;mode&gt;</code>
Shows current latchmode</td></tr>
</table>

Scope	Latchmode command is available in: ▪ P-Series with appl version 2.4.6 or later ▪ A-Series with appl version 1.0.8 or later

5.4.3.7 Graphics

Set graphics on/off

Syntax	`graphics <mode>`
	`graphics`
	graphics command will enable or disable all overlay graphics. When disabled, any key press will enable graphics again.

Param.	**Parameter** / **Type** / **Value** `<mode>` / String / On or off
Query res.	`<mode>` Shows current graphics mode
Scope	graphics command is available in: ▪ P-Series with appl version 2.4.6 or later ▪ A-Series with appl version 1.0.8 or later

5.4.4 Image control commands

Image control command are available in appl mode only.

5.4.4.1 *Palette*

This command selects the current palette and if the palette is to be reversed or not.

Syntax	`palette <filename>` — Select palette. `palette –r <filename>` — Reverse the named palette. `palette –r` — Reverse the current palette. `palette`
Param.	
Query res.	`<name> <state>`

Syntax	Description
`palette <filename>`	Select palette.
`palette –r <filename>`	Reverse the named palette.
`palette –r`	Reverse the current palette.
`palette`	

Parameter	**Type**	**Value**
`<filename>`	String	Name of palette file (e.g. iron.pal)
`<state>`	String	normal, reversed

5.4.4.2 *Level*

Sets the level of the center position in the pixel value span.

Syntax	`level <absolute>` — Absolute level. `level –r <relative>` — Relative current level. `level` — Query (absolute).
Param.	
Query res.	`<absolute>`

Syntax	Description
`level <absolute>`	Absolute level.
`level –r <relative>`	Relative current level.
`level`	Query (absolute).

Parameter	**Type**	**Value**
`<absolute>`	Integer	0 – 65535
`<relative>`	Integer	-65535 – 65535

5.4.4.3 *Span*

Sets the pixel value span.

<table>
<tr><td>Syntax</td><td>
<table>
<tr><td><code>span &lt;absolute&gt;</code></td><td>Absolute gain.</td></tr>
<tr><td><code>span –r <relative&gt;</code></td><td>Relative current span.</td></tr>
<tr><td><code>span</code></td><td>Query (absolute).</td></tr>
<tr><td><code>gain</code></td><td>Equivalent to "text" command. Same syntax.</td></tr>
<tr><td></td><td>"gain" command is obsolete and should not be used</td></tr>
</table>
</td></tr>
<tr><td>Param.</td><td>
<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td><code>&lt;absolute&gt;</code></td><td>Integer</td><td>0 – 65535</td></tr>
<tr><td><code>&lt;relative&gt;</code></td><td>Integer</td><td>-65535 – 65535</td></tr>
</table>
</td></tr>
<tr><td>Query res.</td><td><code>&lt;absolute&gt;</code></td></tr>
</table>

5.4.4.4 Adjust mode

Sets the mode that is used in the automatic adjust of the level and gain settings.

Available modes are level (l), level and span (ls), level+span+histogram (h), level+histogram (lh) or only histogram (oh)

<table>
<tr><td>Syntax</td><td><code>adjmode &lt;mode&gt;</code>
<code>adjmode</code></td></tr>
<tr><td>Param.</td><td>
<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td><code>&lt;mode&gt;</code></td><td>String</td><td>l, ls, h, lh. oh</td></tr>
</table>
</td></tr>
<tr><td>Query res.</td><td><code>&lt;mode&gt;</code></td></tr>
</table>

5.4.4.5 Automatic adjust (histogram)

Starts or stops the automatic adjust of level and gain. Adjust is made from histogram of live image.

<table>
<tr><td>Syntax</td><td><code>autoadj <state&gt;</code>
<code>autoadj.</code></td></tr>
<tr><td>Param.</td><td>
<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td><code>&lt;state&gt;</code></td><td>String</td><td>on, off, now</td></tr>
</table>
where <i>on/off</i> starts/stops the continuous mode and <i>now</i> runs one automatic adjust cycle without altering the current state.
</td></tr>
<tr><td>Query res.</td><td><code>&lt;state&gt;</code></td></tr>
</table>

5.4.4.6 Freeze

This command controls the frame grabber and what the video out circuit shows. Live or the still image just grabbed.

<table>
<tr><td>Syntax</td><td><code>freeze <state&gt;</code>
<code>freeze</code></td></tr>
<tr><td>Param.</td><td><table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td><code>&lt;state&gt;</code></td><td>String</td><td>on, off</td></tr></table></td></tr>
<tr><td>Query res.</td><td><code>&lt;state&gt;</code></td></tr>
</table>

5.4.5 Image file storage

Image file storage commands are available in appl mode only

5.4.5.1 *Store*

This command stores currently displayed image as a file. The file format FFF mentioned below is FLIR Systems propriety file format.

<table>
<tr><td>Syntax</td><td>
<table>
<tr><td><code>store [-j] [-p] [-e] [-o] <file-name&gt;</code></td><td>Store current image as file on current directory.
Default format is FFF with uncompressed IR data.
However, format and compression are depends on options described below.</td></tr>
<tr><td><code>–p</code></td><td>PNG compress the IR data (smallest file).</td></tr>
<tr><td><code>–j</code></td><td>Store image in JPEG format with FFF data in JPEG application tag. File extension will be .jpg no extension is given. May be combined with "-p"</td></tr>
<tr><td><code>-e</code></td><td>Store image in JPEG format without FFF data. File extension will be .jpg no extension is given.
Do not combine with -p</td></tr>
<tr><td><code>-o</code></td><td>No overlay graphics in JPEG frame. Should be combined with -j or -e</td></tr>
</table>
</td></tr>
<tr><td>Param.</td><td>
<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td><code>&lt;filename&gt;</code></td><td>String</td><td>file name</td></tr>
</table>
</td></tr>
<tr><td>Query res.</td><td>No query is defined.</td></tr>
</table>

5.4.5.2 *Recall*

This command recalls a previously stored image from the file system.

<table>
<tr><td>Syntax</td><td>
<table>
<tr><td><code>recall <filename&gt;</code></td><td>Image data (level, gain etc), Palette and IR data is recalled from file.</td></tr>
</table>
</td></tr>
<tr><td>Param.</td><td>
<table>
<tr><td>Parameter</td><td>Type</td><td>Value</td></tr>
<tr><td><code>&lt;filename&gt;</code></td><td>String</td><td>file name</td></tr>
</table>
</td></tr>
<tr><td>Query res.</td><td><code>&lt;name&gt;</code></td></tr>
</table>

5.4.6 Measurement commands

Measurement commands are available in appl mode only

5.4.6.1 *Emiss*

Set emissivity of measured object.

<table>
<tr><td>Syntax</td><td><code>emiss &lt;value&gt;</code>
<code>emiss</code></td></tr>
<tr><td>Param.</td><td><table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td><code>&lt;value&gt;</code></td><td>Float</td><td>0.0-1.0</td></tr></table></td></tr>
<tr><td>Query res.</td><td><code>&lt;value&gt;</code></td></tr>
</table>

5.4.6.2 *Dobj*

Set object distance of measured object.

<table>
<tr><td>Syntax</td><td><code>dobj &lt;value&gt;</code>
<code>dobj</code></td></tr>
<tr><td>Param.</td><td><table><tr><td>Parameter</td><td>Type</td><td>Value</td></tr><tr><td><code>&lt;value&gt;</code></td><td>Float</td><td>Distance in meters 0 – 3047</td></tr></table></td></tr>
<tr><td>Query res.</td><td><code>&lt;value&gt;</code></td></tr>
<tr><td>Scope</td><td>Dobj command is available in:
▪ P-Series with appl version 2.4.6 or later
▪ A-Series with appl version 1.0.8 or later</td></tr>
</table>

5.4.6.3 *Trefl*

Set T reflected for measured object.

<table>
<tr><td>Syntax</td><td><code>trefl &lt;value&gt;</code>
<code>trefl</code></td></tr>
<tr><td>Param.</td><td><table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td><code>&lt;value&gt;</code></td><td>Float</td><td>Temperature in Kelvin</td></tr></table></td></tr>
<tr><td>Query res.</td><td><code>&lt;value&gt;</code></td></tr>
<tr><td>Scope</td><td>Trefl command is available in:
▪ P-Series with appl version 2.4.6 or later
▪ A-Series with appl version 1.0.8 or later</td></tr>
</table>

5.4.6.4 Tatm

Set T atmosphere for measured object.

<table>
<tr><td>Syntax</td><td><code>tatm &lt;value&gt;</code>
<code>tatm</code></td></tr>
<tr><td>Param.</td><td><table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td><code>&lt;value&gt;</code></td><td>Float</td><td>Temperature in kelvin</td></tr></table></td></tr>
<tr><td>Query res.</td><td><code>&lt;value&gt;</code></td></tr>
<tr><td>Scope</td><td>Tatm command is available in:
■ P-Series with appl version 2.4.6 or later
■ A-Series with appl version 1.0.8 or later</td></tr>
</table>

5.4.6.5 Relhum

Set relative humidity for measured object.

<table>
<tr><td>Syntax</td><td><code>relhum <value&gt;</code>
<code>relhum</code></td></tr>
<tr><td>Param.</td><td><table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td><code>&lt;value&gt;</code></td><td>Float</td><td>Relative humidity 0 – 0.99</td></tr></table></td></tr>
<tr><td>Query res.</td><td><code>&lt;value&gt;</code></td></tr>
<tr><td>Scope</td><td>relhum command is available in:
■ P-Series with appl version 2.4.6 or later
■ A-Series with appl version 1.0.8 or later</td></tr>
</table>

5.4.6.6 Measurement box position

Set/Read measurement box position.

<table>
<tr><td>Syntax</td><td><table><tr><td><code>mboxpos <n> <x> <y> &lt;h&gt; &lt;w&gt;</code></td><td>Set mbox position</td></tr><tr><td><code>mboxpos &lt;n&gt;</code></td><td>Read specific mbox position</td></tr><tr><td><code>mboxpos</code></td><td>Read all mbox positions</td></tr></table></td></tr>
</table>

Param.

Parameter	Type	Value
<n>	Integer	Measurement box number (1..)
<x>	Integer	Horizontal position
<y>	Integer	Vertical position
<w>	Integer	Horizontal size
<h>	Integer	Vertical size

Query res.

```
<n> <x> <y> <h> <w>
```

5.4.6.7 Measurement box signal values

Read measurement box signal values.

Syntax

mboxsig <n>	Read specific mbox value
mboxsig	Read all mbox values

Param.

Parameter	Type	Value
<n>	Integer	Measurement box number (1..)
<maxsig>	Float	Max signal value within mbox
<minsig>	Float	Min signal value within mbox
<avgsig>	Float	Average signal value within mbox
<stddev>	Float	Standard deviation in signal value
<xmax>	Integer	Horizontal position for maxsig
<ymax>	Integer	Vertical position for maxsig
<xmin>	Integer	Horizontal position for minsig
<ymin>	Integer	Vertical position for minsig

Query res.

```
mboxsig <n> <maxsig> <minsig> <avgsig> <stddev>
(<xmax>,<ymax>) (<xmin>,<ymin>)
```

5.4.6.8 Measurement box temperature

Read measurement box temperatures.

Syntax

`mboxtemp <n>`	Read specific mbox temperature
`mboxtemp`	Read all mbox temperatures

Param.

Parameter	**Type**	**Value**
`<n>`	Integer	Measurement box number (1..)
`<maxtemp>`	Float	Max temperature in kelvin within mbox
`<mintemp>`	Float	Min temperature in kelvin within mbox
`<avgtemp>`	Float	Average temperature in kelvin within mbox
`<stddev>`	Float	Standard deviation in kelvin
`<c>`	Char	A character showing calibration status for corresponding max/min/avg/stddev = Value is within calibrated temprange limits < Value is below measurable limits > Value is above measurable limits * Value is uncertain (outside calibrated temprange, but still possible to show a temperature) - Impossible to show temperature
`<xmax>`	Integer	Horizontal position for maxtemp
`<ymax>`	Integer	Vertical position for maxtemp
`<xmin>`	Integer	Horizontal position for mintemp
`<ymin>`	Integer	Vertical position for mintemp

Query res.

```
mboxtemp <n> <maxtemp> <mintemp> <avgtemp> <stddev>
<c><c><c><c> (<x>,<y>) (<x>,<y>)
```

5.4.6.9 Measurement functions

Read/set measurement functions activity state

Syntax

`mfunc <type> <id> <state>`	Set measurement function state
`mfunc`	Read all measurement function state
`mfunc <type>`	Read all measurement function state of `<type>`
`mfunc <type> <id>`	Read specific measurement function state

Param.

Parameter	**Type**	**Value**
`<type>`	Sting	spot, mbox, isotherm
`<id>`	Integer	spot/mbox.. id number (1..)
`<state>`	Integer	on, off

Command currently does not implement `<type>` "circle" or "line"

Query res. `mfunc <type> <id> <state>`

5.4.6.10 Spotmeter position

Read/set spotmeterposition.

Syntax

`spotpos <n> <x> <y>`	Set spotmeter position
`spotpos <n>`	Read specific spotmeter position
`spotpos`	Read all spotmeter positions

Param.

Parameter	**Type**	**Value**
`<n>`	Integer	Spotmeter number (1..)
`<x>`	Integer	Horizontal position in pixels
`<y>`	Integer	Vertical position in pixels

Query res. `spotpos <n> <x> <y>`

5.4.6.11 Spotmeter value

Read spotmeter value.

Syntax

`spotsig <n>`	Read specific spotmeter value
`spotsig`	Read all spotmeter values

Param.	<table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td>`<n>`</td><td>Integer</td><td>Spotmeter number (1..)</td></tr><tr><td>`<value>`</td><td>Float</td><td>Signal value</td></tr></table>
Query res.	`spotsig <n> <value>`

5.4.6.12 Spotmeter temperature

Read spotmeter temperature.

Syntax	<table><tr><td>`spottemp <n>`</td><td>Read specific spotmeter temperature</td></tr><tr><td>`spottemp`</td><td>Read all spotmeter temperatures</td></tr></table>
Param.	<table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td>`<n>`</td><td>Integer</td><td>Spotmeter number (1..)</td></tr><tr><td>`<value>`</td><td>Float</td><td>Temperature in kelvin</td></tr><tr><td>`<c>`</td><td>Char</td><td>A character showing calibration status for <value> See mboxtemp...</td></tr></table>
Query res.	`spottemp <n> <value> <c>`

5.4.7 File system commands

5.4.7.1 Verify

Verify integrity of file system and repair any found errors if possible.

Syntax	`verify`
Param.	No parameters.
Query res.	Disk status and found errors reported. Response is intended for human reading.

5.4.7.2 Print working volume

Get working volume number.

<table>
<tr><td>Syntax</td><td><code>pwv</code></td></tr>
<tr><td>Param.</td><td><table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td>&lt;volume&gt;</td><td>String</td><td>Volume name (e.g. 28.0)</td></tr></table></td></tr>
<tr><td>Query res.</td><td><code>&lt; volume &gt;</code></td></tr>
</table>

5.4.7.3 Check free file system space

Report free file system space

<table>
<tr><td>Syntax</td><td><code>df</code></td></tr>
<tr><td>Param.</td><td>No parameters.</td></tr>
<tr><td>Query res.</td><td>Disk free status is presented. One row per volume. Output is intended for human reading
Example:<table><tr><td>filesystem</td><td>kbytes</td><td>used</td><td>avail</td><td>usage%</td><td>mounted</td></tr><tr><td>28.0/</td><td>14561280</td><td>7430144</td><td>7131136</td><td>51.0%</td><td>OK</td></tr><tr><td>3.0/</td><td>409600</td><td>36864</td><td>372736</td><td>9.0%</td><td>OK</td></tr></table></td></tr>
<tr><td>Scope</td><td>df command is available in appl mode only</td></tr>
</table>

5.4.8 Other commands

5.4.8.1 Version

This command returns the version and ID of all hardware and software of the system.

Syntax:

version

Parameters:

Parameter	Type	Value
`<sw item>`	String	camera, optics, boot2, appl etc
`<version>`	String	Major.Intermediate.Minor (e.g. 1.0.0)
`<edit>`	String	Edit number
`<date>`	String	Build date (e.g. 21-Jun-2001)
`<build type>`	String	Build type. Identifies type of software; Release, TRelease
`<hw item>`	String	board name/type
`<part no>`	String	Part number
`<serial>`	String	Serial number of board
`<board revision>`	String	Board revision number

Query result:

```
<sw item> <version> <edit> <date> <build type>

<hw item> <part no> <serial> <board revision>
```

One row per item.

Example:

OK>vers				
ThermoVision A40	2369900000	23600002	01	2003-08-20
naco	1195409	1114 04		

optionboard	1195452	1091 02		
ioboard	1195412	1170 04		
focusboard	1195726	1003 01		
optics	1195205	2432 05		
myst		001900		
riven		003896		
fpa		029670		
boot1	1.2.10	1	24-Jan-2003	(Release)
boot2	1.0.6	1	10-Nov-2003	(Release)
appl	1.0.6	44	18-Nov-2003	(TRelease)
filekit	1.2.0T	-	18-Nov-2003	AutMn
prodkit	1.2.0	-	10-Nov-2003	A40 prod
webkit	1.2.0	-	10-Nov-2003	A20/40 web
config	A40M	1.0	-	
language	I_1.0.6	44	18-Nov-2003	(TRelease)
fpga	NACOA_C 2.0	0	11-Jun-2003	
pic	3.1	-	-	-
detfpga	2.0.54	0	-	-
focuspic	5.5	-	-	-
optfpga	NAXFA_D 1.3	0	24-Jun-2003	
calib	1.3-MR2	Def_a	-	-
OK>				

Scope:

version command is available in appl mode only

5.4.8.2 Test image

Displays a test image on the screen. The test image is generated by the FPGA on the MACT/NECC board.

<table>
<tr><td>Syntax</td><td><code>videotest <state&gt;</code></td></tr>
<tr><td>Param.</td><td>
<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td><code>&lt;state&gt;</code></td><td>String</td><td>on, off</td></tr>
</table>
</td></tr>
<tr><td>Query res.</td><td>No query defined.</td></tr>
<tr><td>Scope</td><td>videotest command is available in appl mode only</td></tr>
</table>

5.4.8.3 *Button simulation command*

This command could be used for a simple remote control. It simulates press/release of buttons on the camera used for direct control.

<table>
<tr><td>Syntax</td><td><code>bt [&lt;press option>] <button&gt;</code></td></tr>
<tr><td>Param.</td><td>
<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td><code>&lt;-press option&gt;</code></td><td>String</td><td>“-p” – simulate a press
“-r” – Simulate a release</td></tr>
<tr><td><code>&lt;button&gt;</code></td><td>String</td><td>“e”/”E” – Enter/YES button
”c”/”C” – Cancel/NO button
“t”/”T” – SEL button
”s”/”S” – SAVE/FRZ button
”a”/”A” – A button
“g”/”G” – Laser button
“o”/”O” – On/Off button (CMpro)
”l”/”L” – Left arrow button
”r”/”R” – Right arrow button
”u”/”U” – Up arrow button
”d”/”D” – Down arrow button
“1”-“4” – F1-F4 function key</td></tr>
</table>
Omitted “–p”/”-r” simulates a press/release of <button>
Long press of button is simulated using several “bt –p <button>” followed by a “bt –r <button>”
</td></tr>
<tr><td>Scope</td><td>bt command is available in appl mode only</td></tr>
</table>

5.5 pSH+ Built-In Commands

This appendix is a cut and paste directly from “pSOSystem Concepts and User’s guide” chapter 11.9.6 - “pSH+ Built-In Commands”.

The original document is published by WindRiver Systems Inc.

5.5.1 Command summary

arp	Address resolution display and control.
cat	Concatenate and display files.
cd	Change working directory.
clear	Clear the terminal screen.
cmp	Perform a byte-by-byte comparison of two files.
cp	Copy files.
date	Display or set the date.
du	Display disk blocks usage.
echo	Echo arguments to the standard output.
getid	Get NFS user ID and group ID. (see **setid**)
getpri	Get task priority. (see **setpri**)
head	Display the first few lines of the specified files.
help	Display reference manual pages.
history	Display a previous command, using the command number.
ifconfig	Configure network interface parameters.
kill	Terminate a task.
ls	List the contents of a directory.
mkdir	Create a directory.
mkfs	Construct a pHILE+ file system.
mount	Mount a pHILE+ file system.
mv	Move or rename files.
netstat	Show network status.
nfsmount	Mount a NFS file system.

pcmkfs	Construct an MS-DOS file system.
pcmount	Mount an MS-DOS file system.
ping	Send ICMP ECHO REQUEST packets to network hosts.
popd	Pop the directory stack.
pushd	Push current directory onto the directory stack.
pwd	Display pathname of the current working directory.
resume	Resume a task.
rm	Remove files.
rmdir	Remove directories.
route	Manually manipulate the routing tables.
setenv	Set environment variables.
setid	Set NFS user ID and group ID.
setpri	Set task priority.
sleep	Suspend execution for a specified interval.
suspend	Suspend a task.
sync	Force all changed blocks to disk.
tail	Display the last part of a file.
touch	Update the modification time of a file.
umount	Unmount a file system.

5.5.2 Detailed description of commands

arp *hostname*

arp -a

arp -d *hostname*

arp -f *filename*

arp -s *hostname ether_address* [**temp**] [**pub**]

Address resolution display and control. The **arp** command displays and modifies the Internet-to-Ethernet address translation tables used by the address resolution protocol.

With no flags, the program displays the current ARP entry for *hostname*. The host may be specified by number, using Internet dot notation.

The **arp** options are as follows:

-a	Display the current ARP entries.
-d	Delete an entry for the host called *hostname*.
-f	Read the file named filename and set multiple entries in the ARP tables. Entries in the file should be of the form *hostname ether_address* [**temp**] [**pub**] (see option **-s** for argument definitions).
-s	Create an ARP entry for the host called *hostname* with the Ethernet address ether_address. The Ethernet address is given as six hexadecimal bytes separated by colons. The entry will be permanent unless the word temp is given in the command. If the word pub is given, the entry will be published. For instance, this system will respond to ARP requests for *hostname* even though the *hostname* is not its own. **arp -s** can be used for a limited form of proxy ARP when a host on one of the directly attached networks is not physically present on the subnet. Another machine can then be configured to respond to ARP requests using arp -s. This is useful in certain SLIP or PPP configurations.

cat [**-benstv**] [*filename* ...]

Concatenate and display. **cat** sequentially reads each *filename* and displays the contents of each named file on the standard output.The following input displays the contents of **goodies** on the standard output:

```
psh> cat goodies
```

Note that **cat** does not redirect the output of a file to the same file. For example,

cat fails for filename1 > filename1 or filename1 >> filename1.

You should avoid this type of operation, because it can cause the system to go into an indeterminate state. **cat** options are as follows:

b	Number the lines, but omit the line numbers from blank lines (similar to **-n**).
e	Display non-printing characters, and additionally display a $ character at the end of each line (similar to **-v**).
n	Precede each line output with its line number.
s	Substitute a single blank line for multiple adjacent blank lines.

t	Display non-printing characters (like the **-v** option), and additionally display [TAB] characters as **^I** (a [CTRL]-I).
v	Display non-printing characters (with the exception of [TAB] and [NEWLINE] characters), so they are visible. Control characters print like **^X** (for [CTRL]-X); the [DEL] character (octal 0177) prints as **^?**. Non-ASCII characters (with the high bit set) are displayed as **M-x** where **M-** stands for "meta" and **x** is the character specified by the seven loworder bits.

cd [*directory*] Change working directory. The argument directory becomes the new working directory.

clear

Clear the terminal screen (if possible). Set the terminal type and the clear sequence to clear the screen. Few standard terminaltypes are supported in the sample application. Customers can

add support for other terminal types.

cmp [**-ls**] filename1 filename2 [skip1] [skip2]

Perform a byte-by-byte comparison of *filename1* and *filename2*. With no arguments, **cmp** makes no comment if the files are the same. If they differ, it reports the byte and line number at which differences occur, or else it reports that one file is an initial subsequence of the other. Arguments *skip1* and *skip2* are initial

byte offsets into *filename1* and *filename2*, respectively, and can be either octal or decimal (a leading zero denotes octal).

cmp options are:

l	Silent. Print nothing for differing files.
s	Silent. Print nothing for differing files.

cp [**-i**] *filename1 filename2*

cp -rR [**-i**] *directory1 directory2*

cp [**-irR**] *filename ... directory*

On the first line of the synopsis, the **cp** command copies the contents of *filename1* to *filename2*. If *filename1* is a symbolic link or a duplicate hard link, the contents of the file that the link refers to are copied, but the links are not preserved. On the second line of the synopsis, **cp** recursively copies *directory1* along with its contents and subdirectories to *directory2*.

If *directory2* does not exist, **cp** creates it and duplicates the files and subdirectories of *directory1* within it.

If *directory2* exists, **cp** copies *directory1* (as a subdirectory) within *directory2*, along with its files and subdirectories.

On the third line of the synopsis, each *filename* is copied to the indicated directory. The basename of the copy corresponds to that of the original. The destination directory must already exist for the copy to succeed.

cp does not copy a file to itself.

cp options are as follows:

i	Interactive: a prompt for confirmation of the copy appears whenever the copy would overwrite an existing file. A **y** answer confirms that the copy should proceed. Any other answer prevents **cp** from overwriting the file.
r	See **R**.
cp R	Recursive. If any of the source files are directories, copy the directory along with its files (including any subdirectories and their files). The destination must be a directory.In the following example, the first command line entry starts the copy operation. The second command line lists the contents of the directory to verify the results of the copy.

To copy a file:

```
psh> cp goodies goodies.old

psh> ls

goodies goodies.old
```

To copy a directory, first to a new and then to an existing directory, enter the following:

```
psh> cp -r src bkup

psh> ls -R bkup

x.c y.c z.sh

psh> cp -r src bkup

psh> ls -R bkup

src x.c y.c z.sh

src:
```

```
x.c y.c z.sh
```

date *[yyyymmddhhmm [.ss]]*

Without an argument, **date** displays the current date and time. Otherwise, it sets the current date according to the input argument.

The argument part *yyyy* is the four digits of the year; the first *mm* is the month number; *dd* is the day number in the month; *hh* is the hour number (24 hour system); the second *mm* is the minute number; and *.ss* (optional) specifies seconds. If *yyyy* is the current year, it can be omitted as the current year value is the default.

To set the date to Oct 8, 12:45 AM, type:

```
date 10080045
```

du [**-sa**] [*filename* ...]

Display the number of 512-byte disk blocks used per file or directory. This command can display the block count of one or more specified files; all files in either the current or another specified directory; or, recursively, the number of blocks in directories within each specified directory. If no *filename* is given, the current directory (symbolized by a.) is used. Filenames can contain wildcards.

du options are as follows:

s	Display only the total for each of the specified filenames.
a	Generate an entry for each file.
	Entries are generated only for each directory in the absence of options.

The following is an example of **du** usage in a directory. The example uses the **pwd** command to identify the directory, then uses **du** to show the usage of all the subdirectories in

that directory. The total number of blocks in the directory (1211) is the last entry in the display:

```
psh> pwd

/junk

psh> du

5 ./junk1
```

```
33 ./xxxxx

44 ./vvvvv/vvvv.junk1

217 ./vvvvv/vvvv.junk2

401 ./vvvvv

144 ./mmmmm

80 ./gggggg

388 ./ffffff

93 ./mine

15 ./yours

1211 .
```

echo [-n] [*argument* ...]

Echo *argument*(s) to the standard output. Arguments must be separated by [SPACE] characters or [TAB] characters and terminated by a [NEWLINE].

The **-n** option keeps a [NEWLINE] from being added to the output.

getid Get the user ID and group ID of the shell task. For example:

```
psh> getid

uid: 23, gid: 140
```

where the second line is output displayed on standard output.

getpri tname|-tid Return the priority of a task, specified by either the task name

(**tname**) or the task identifier (**tid**). For example:

```
psh> getpri ROOT

ROOT task priority = 250
```

head [-n] *filename*... Copy the first **n** lines of each *filename* to the standard output. The default value of **n** is 10.

When more than one file is specified, the start of each file appears as follows:

```
==>filename<==
```

For example, the following line

psh> **head -4 junk1 junk2**

produces

```
==> junk1 <==

This is junk file one

==> junk2 <==

This is junk file two
```

help [*command_name*]

Print information about shell commands to the console. If a valid command name is given, **help** prints out information about that command. With no command name for an input, **help** prints out a list of shell commands.

The following example shows the results of **help** without an argument:

psh> **help**

help cat cmp echo help mkfs pcmkfs pushd rmdir sleep cd cp getid kill mount pcmount pwd setenv suspend clear date getpri ls mv ping resume setid sync console du head mkdir nfsmount popd rm setpri

The following example shows the result of **help cat**.

psh> **help cat**

```
cat - concatenate and display (reentrant, not locked)
```

history Displays a list of previously executed commands on this shell instance. The commands to be displayed are specified by command number in the history command.

The **history_len** parameter specifies the number of commands stored in the history buffer. Setting **history_len** to **0** disables the history feature.

Commands can be recalled by using the following ‘csh’ syntax:

!*cmd - number*

Where *cmd-number* is the previously executed command that will be displayed by the history command.

During session creation, memory is pre-allocated for storing the history commands. 256 bytes are allocated for each command in history.

ifconfig interface_number [address_family] [address [dest_address]] [**up**] [**down**]

[**netmask** *mask*] [**broadcast** *address*] [**-a**] [**arp** | **-arp**]

Configure network interface parameters. The **ifconfig** command is used to assign an address to a network interface and/or to configure network interface parameters. It may be used to redefine an interface's address or other operating parameters.

The *interface_number* parameter is the interface number of the network interface in pNA+ (Wind River TCP/IP stack in pSOS+).

The only *address_family* supported is the TCP/IP family (that is, inet).

For the TCP/IP family (inet), the *address* is a TCP/IP address expressed in the Internet standard dot notation. Typically, an Internet address specified in dot notation will consist of the system's network number and the machine's unique host number.

ifconfig For the ether address family, the *address* is an Ethernet address

represented as x:x:x:x:x:x where x is a hexadecimal number between 0 and FF.

If the *dest_address* parameter is supplied in addition to the address parameter, it specifies the address of the correspondent on the other end of a point to point link.

up	Mark an interface up. This happens automatically when setting the first address on an interface. The up option enables an interface after an **ifconfig down**, reinitializing the hardware.
down	Mark an interface down. When an interface is marked down, the system will not attempt to transmit messages through that interface. If possible, the interface will be reset to disable reception as well. This action does not automatically disable routes using the interface.
broadcast	*address* Specify the address to use to represent broadcasts to the network. The default address is the address with a host part of all 1's.

netmask	*mask* Specify how much of the address to reserve for subdividing networks into sub-networks. The mask includes the network part of the local address and the subnet part, which is taken from the host field ofthe address. The mask contains 1's for the bit positions in the 32-bit address that are to be used for the network and subnet parts, and 0's for the host part. The mask should contain at least the standard network portion, and the subnet field should be contiguous with the network portion. The mask can be specified: n with a single hexadecimal number with a leading 0x. n with a dot-notation address.

The **ifconfig** options are as follows:

-a	Display the current configuration for all network interfaces in the system.
arp	Enable the use of the Address Resolution Protocol in mapping between network level addresses and link level addresses (default). This is currently implemented for mapping between TCP/IP addresses and 10Mb/s Ethernet addresses.
-arp	Disable the use of the Address Resolution Protocol.

kill *tname|-tid*

Terminate the task indicated by either the task name (*tname*) or the task identifier (**tid**). The **kill** command does this by calling **t_restart** with a second argument of **-1**. The task must be designed to read this second argument and do its own resource cleanup, then terminate. For example:

```
psh> kill tftd
```

ls [-aACdfFgilqrRs1] *filename ...*

For each filename that is a directory, **ls** lists the contents of the directory; for each filename that is a file, **ls** repeats its name and any other information requested. By default, the output is sorted

alphabetically. **ls** options are as follows:

a	
A	List all entries.
(**ls** only) Same as **-a**, except that the . and the .. are not listed.	
C	Force multi-column output, with entries sorted down the columns; for **ls**, this is the default when output goes to a terminal.
d	If argument is a directory, list only its name (not its contents); often used with **-l** to get the status of a directory.

f	Force each argument to be interpreted as a directory and list the name found in each slot. This option turns off **-l**, **-s**, and **-r** and turns on **-a**; the order is the same as the order of the entries appearing in the directory.
F	Mark directories with a trailing slash */ and executable files with a trailing asterisk (*).
g	Shows group ownership of the file in a long output.
i	For each file, print the **i**-number in the first column of the report.
l	List in long format. Long format shows the mode, the number of links, the owner, the size (in bytes), and the time of each file's last modification. If the last modification occurred more than six months ago, the display format is **month-date-year**; the format for files modified in six or less months is **month-date-time**.
ls q	Display nongraphic characters in filenames as the ? character; for **ls**, this is the default when output goes to a terminal.
r	Reverse the order of the sort either to reverse the alphabetic order or list the oldest data first.
R	Recursively list subdirectories encountered.
s	Give size of each file. Include indirect blocks used to map the file. Display in Kbytes.
l	Force single-column output.

mkdir [-p] *dirname...*

Create a directory. The **-p** option allows missing parent directories to be created, as needed. For example:

```
psh> ls -lR

total 8

-r--r--r-- 1 root 512 Mar 31 94 00:00 BITMAP.SYS

-r--r--r-- 1 root 2048 Mar 31 94 10:01 FLIST.SYS

drwxrwxrwx 1 root 32 Mar 31 94 13:34 test_dir

./test_dir:

psh> mkdir -p new_dir/next_dir
psh> ls -lR

total 9

-r--r--r-- 1 root 512 Mar 31 94 00:00 BITMAP.SYS

-r--r--r-- 1 root 2048 Mar 31 94 10:01 FLIST.SYS

drwxrwxrwx 1 root 16 Mar 31 94 13:36 new_dir

drwxrwxrwx 1 root 32 Mar 31 94 13:34 test_dir

./new_dir:

total 0
```

mkfs [**-i**] *volume_name label size num_of_fds*

Initialize a file system **volume_name** and label it with *label*. The argument *size* is the volume size, and *num_of_fds* is the number of file descriptors.

The **-i** option initializes a device driver for the device. For example:

```
psh> mkfs 5.6 HDSK 2096 512

Warning: this operation will destroy all data on the

specified volume.

Do you want to continue (y/n)? y

psh>
```

mount *volume_name* [*sync_mode*]

Mount a pHILE+ formatted volume on the file system. (A volume must be mounted before any file operations can be executed on it.) Permanent (non-removable media) volumes need to be mounted only once. Removable volumes must be mounted and unmounted as required. The *sync_mode* is one of the following:

0	Specifies immediate-write synchronization mode

1	Specifies control-write synchronization mode.
2	Specifies delayed-write synchronization mode (the default).

For example:

```
psh> mount 5.6/
```

mv [-if] *filename1 filename2*

mv [-if] *directory1 directory2*

mv [-if] *filename... directory*

Move around files and directories in the file system. A side effect of **mv** is that it renames a file or a directory. The three major forms of **mv** appear in the preceding synopses.

The first form of **mv** moves (and changes the name of) *filename1* to *filename2*. If *filename2* already exists, it is removed before *filename1* is moved.

The second form of **mv** moves (and changes the name of) *directory1* to *directory2* but only if *directory2* does not already exist. If *directory2* exists, the third form applies.

The third form of **mv** moves one or more filenames (can also be directories) with their original names into the last directory in the list. **mv** does not move either a file to itself or a directory to itself. **mv** options are as follows:

i	Interactive mode. **mv** displays the name of the file followed by a question mark whenever a move would replace an existing file. If a line starts with **y**, **mv** moves the specified file; otherwise, **mv** does nothing with the file.
f	Force. Override any mode restrictions and the **i** option.

netstat [-airs]

netstat displays the contents of various network-related data structures in various formats. **netstat** with no option will display all sockets other than the ones related to server tasks.

netstat options are as follows:

a	Show the state of all sockets including ones that are listening (server tasks).
i	Show the state of all network interfaces.
r	Show the routing tables.
s	Show per-protocol statistics.

nfsmount host: host_directory directory

Mount the remote file system using NFS protocol. The host **host** should advertise the directory, *host_directory* for this command to complete successfully. The **host** argument can be either an IP address or a hostname if the Name Resolver is configured.

pcmkfs [-i] *volume_name size*

Do a **pcinit_vol** of the volume **volume_name** for the disk type **size**, where **size** is one of the following:

1	360 Kbyte (5 1/4" double density)
2	1.2 Mbyte (5 1/4" high density)
3	720 Kbyte (3 1/2" double density)
4	1.4 Mbyte (3 1/2" high density)

The **-i** option initializes the device. For example:

```
psh> pcmkfs 5.3 4

Warning: this operation will destroy all data on the

specified volume.

Do you want to continue (y/n)? y
```

pcmount *volume_name* [*sync_mode*]

Mount an MS-DOS file system *volume_name*. (A volume must be mounted before any file operations can be executed on it.) The argument *sync_mode* can be one of the following:

0	Immediate write synchronization mode.

1	Control write synchronization mode.
2	Delayed write synchronization mode (default).

For example:

```
psh> pcmount 5.3
```

ping [**-s**] *host* [*timeout*]

The **ping** command uses the ICMP protocol's mandatory

ECHO_REQUEST datagram to elicit an ICMP

ECHO_RESPONSE from the specified host or network gateway.

ECHO_REQUEST datagrams (pings) have an IP and ICMP header followed by a **struct timeval** and then an arbitrary number of bytes to pad out the packet. If the host responds, **ping** prints **host is alive** on the standard output and exits. Otherwise, after *timeout* seconds, it writes **no answer from host**. The default value of *timeout* is 10.

When the s option is specified, **ping** sends one datagram per second and prints one line of output for every **ECHO_RESPONSE** that it receives. No output is produced if no response occurs. The default size for a datagram packet is 64 bytes. The **host** argument can be either an IP address or a hostname if the Name Resolver is configured.

When using **ping** for fault isolation, first **ping** the local host to verify that the local network interface is running.

For example:

```
psh> ping 192.103.54.190

PING (192.103.54.190): 56 data bytes

192.103.54.190 is alive
```

popd

Pop the directory stack and change to the new top directory. For example:

```
psh> pushd test_dir

psh> pwd

5.5/test_dir

psh> popd

psh> pwd

5.5/
```

pushd *directory*

Push the current *directory* onto the directory stack and change the current working directory to that directory. For example:

```
psh> pwd

5.5/

psh> pushd test_dir

psh> pwd

5.5/test_dir
```

pwd

Display the pathname of the current working directory. For example:

```
psh> cd 5.5//usr

psh> pwd

5.5//usr
```

resume *tname* **|** **-***tid*

Resume a suspended task by the task name (*tname*) or the task identifier *tid*. For example:

psh> resume ROOT

rm [-fir] *filename* ...

Remove (unlink directory entries for) one or more files. If an entry was the last link to the file, the contents of that file are lost. **rm** options are as follows:

f	Force removal of files without displaying permissions or questions and without reporting errors.
i	Prompt whether to delete each file and, under **-r**, whether to examine each directory. (This is sometimes called the interactive option.)
r	Recursively delete the contents of a directory, its subdirectories, and the directory itself.

rm Examples:

```
psh> ls -lR

total 9

-r--r--r-- 1 root 512 Mar 31 94 00:00 BITMAP.SYS

-r--r--r-- 1 root 2048 Mar 31 94 10:01 FLIST.SYS

drwxrwxrwx 1 root 16 Mar 31 94 13:36 new_dir

drwxrwxrwx 1 root 32 Mar 31 94 13:34 test_dir

./new_dir:

total 0

drwxrwxrwx 1 root 0 Mar 31 94 00:00 next_dir

./new_dir/next_dir:

./test_dir:

total 1

-rwxrwxrwx 1 root 33 Mar 31 94 00:00 test_file
```

```
psh> rm -rf new_dir

psh> ls -lR

total 8

-r--r--r-- 1 root 512 Mar 31 94 00:00 BITMAP.SYS

-r--r--r-- 1 root 2048 Mar 31 94 10:01 FLIST.SYS

drwxrwxrwx 1 root 32 Mar 31 94 13:34 test_dir

./test_dir:

total 1

-rwxrwxrwx 1 root 33 Mar 31 94 00:00 test_file
```

rmdir *directory ...*

Remove each named directory. **rmdir** removes only empty directories.

route [**-a**] **add** | **delete** [**host** | **net**] *destination* [*gateway* [*metric*]]

This command manually manipulates the network routing tables normally maintained by the system routing daemon, routed, or through default routes and redirect messages from routers.

route allows the pSH user to operate directly on the routing table for the specific host or network indicated by *destination*. Default is available for gateways to use after all other routes have been attempted. The *gateway* argument, if present, indicates the network gateway to which packets should be addressed. The **metric** argument indicates the number of hops to the destination. The metric is required for *add* commands; it must be 0 if the destination is on a directly attached network, and nonzero if the route utilizes one or more gateways.

add	command instructs **route** to add a route to *destination*.
delete	deletes a route.

Routes to a particular host must be distinguished from those to a network. The optional keywords **net** and **host** force the destination to be interpreted as a network or a host, respectively. Otherwise, if the destination has a local address part of **INADDR_ANY**, then the route is assumed to be to a network; otherwise, it is presumed to be a route to a host. If the route is to a destination connected by a gateway, the metric parameter should be greater than 0. All addresses should be specified in standard IP address dot notation.

The **route** option is as follows:

-a	Display the routing table.

setenv *variable_name value*

Change a pSH+ environment *variable_name* to a new *value*. If used without arguments, **setenv** prints a list of pSH+ variables and their values.

Note that the only variable that can be changed is **TERM**.

Examples:

```
psh> setenv

CVOL=5.5

CDIR=/

SOFLIST=5

LOGNAME=guest

IND=0

OUTD=0

TERM=sun

psh> setenv TERM vt100

psh> setenv

CVOL=5.5

CDIR=/

SOFLIST=5
```

```
LOGNAME=guest

IND=0

OUTD=0

TERM=vt100
```

setid uid gid

Change the *uid* and *gid* ID of the current pSH+ session. For example:

```
psh> getid

uid: 23, gid: 140

psh> setid 2 3

psh> getid

uid: 2, gid: 3
```

setpri tname **|** -tid new_priority

Set the *new_priority* of the task identified by either the task name (*tname*) or task identifier (*tid*). For example:

```
psh> getpri ROOT

ROOT task priority = 76

psh> setpri ROOT 252

psh> getpri ROOT

ROOT task priority = 252
```

sleep *time*

Suspend execution for the number of seconds specified by *time*.

suspend *tname* **|** *-tid*

Suspend the task identified by either the task name (*tname*) or the task identifier (*tid*). For example:

psh> suspend tnpd

sync

Update a mounted volume by writing to the volume all modified file information for open files and cache buffers that contain modified physical blocks. This call is superfluous under immediate-write synchronization mode and is not allowed on an NFS volume. For example:

```
psh> sync
```

tail + | *-number* [**lc**] *filename*

Copy *filename* to standard output beginning at a designated place.

tail options are typed contiguously and are not separated by dashes (-). The options are as follows:

tail +*number*	Begin copying at distance *number* from the **beginning** of the file. *number* is counted in units of lines or characters, according to the appended option **l** or **c**. If no units are specified, counting is by lines. If *number* is not specified, 10 is used.
-number	Begin copying at distance *number* from the **end** of the file. *number* is counted in units of lines or characters, according to the appended option **l** or **c**. If no units are specified, counting is by lines. If *number* is not specified, 10 is used. l *number* is counted in units of lines.
c *number*	is counted in units of characters.

touch [**-cf**] *filename...*

Set access and modification times of each argument to the current time. A file is created if it does not already exist. The options are:

c	Do not create file if it does not already exist.
f	Attempt to force the touch regardless of read and write permissions on *filename*.

umount *directory*

Unmount a previously mounted file system where *directory* is the mount point of the file system. The file system is synchronized (all memory-resident data is flushed to the device). For example:

```
psh> mount 5.6
psh> cd 5.6/
psh> ls
BITMAP.SYS FLIST.SYS
psh> cd 5.5/
psh> umount 5.6
psh> cd 5.6/
5.6/: no such file or directory
```

6 Common resources

6.1 Introduction

This section describes the resources supported by FLIR Systems for end-customer use. They should be stable enough over the product lifetime, although no absolute guarantee could be given.

6.1.1 General information

A lot of (but not all) software functionality within the A-, P/S- and E-series cameras is exposed through something called “software resources”. Those that are familiar with the Microsoft® Windows registry will recognize the idea.

However, in the A-series cameras (as well as in E-/P-/S-Series) a resource node could also represent a software function that upon read or write actively interacts with the software system.

Resources are organized in a tree hierarcy:

- A resource node could be a data holder only (for instance camera calibration)
- A resource node could be connected to a software function. (for instance temp sensor resources)
- A resource node has a type (for instance double, int32, ascii) and certain attributes (readonly, read/write)

6.1.2 Non-documented resources

10474103;1

There are also a number of non-documented resources. Non-documented resources are for FLIR internal usage only. Their interpretation and/or existence may change from one software version to another without any written notice, and end-user are strongly discouraged to trying to use any non-documented resources even if found.

6.1.3 Indirect resource access

The camera resources are read/manipulated indirectly by the different interfaces to the camera.

For instance the grapical user interface (buttons and video overlay) manipulates the system by reading/writing resources in code. The web browser interface interacts with resourses by using asp files (within the camera file system) that translates resource names to presentable values upon access. URL's sent to the camera by clicking on links could also manipulate resources.

6.1.4 Direct resource access

It is also possible to read/manipulate the resources directly.

- The SDK camera control has a hook to allow direct resource read/write. For more information, see **GetCameraProperty** (section 6.3.8 on page 52) and **SetCameraProperty** methods (section 6.3.14 on page 63) in Publ. No. 1 557 738 ThermoVision SDK User's Manual.
- There is a FLIR tcp socket server interface (tcp/ip port 0x5678).
- Through special (protected) commands rls, rset.
  These commands are however not intended for the end user. The reason for not allowing end user access here is to discourage possible user development of non forward compatible applications, because the command set reveals also not documented resources that may change in later releases of the camera software.

6.1.5 Notation

The resource paths are truncated, and each previous level is reduced to a dot. Example: `.image.adj.level` is listed as `...level`.

Attribute	Explanation	Comment
r	read allowed	(**r**) means the resource always reads the same, or the read value is meaningless
w	write allowed	**(w)** means that write only is to be performed by task owning resource
c	create allowed	
d	delete allowed	
p	persistent (will be stored in the registry file)	(**p**) means the persistence flag may or may not be set

Attribute	Explanation	Comment
s	subscription allowed	
n	notify parent's subscribers	

6.1.6 Compatibility

Notation in document	Camera series
M	Including T, C, F, A (below)
T	ThermaCAM E series
C	ThermaCAm P/S series with BAe detectors
F	ThermaCAM P/S series with ULIS detectors
A	ThermoVisionn A20 and A40 series

NOTE: Common resources are not marked.

6.2 Resource list

6.2.1 .image

Name	Type	Attr.	com	Description; allowed values
`.image`	entry	-		
`..videomode`	ascii	rw		”NTSC”, ”PAL”
`..adj`	entry	-		**SEE ALSO:** For more information, see section 6.2.1.1 – .image.adj on page 80
`..corr`	entry	-		
`...nuc`	bool	(r)wc		Perform non-uniform correction; true
`...nucFramesLog2`	int32	rwsp		Log2 of num. frames to average in NUC; 1–7
`...nucShutter`	bool	rw		NUC uses shutter; true/false

Name	Type	Attr.	com	Description; allowed values
`..file`	entry	-		**SEE ALSO:** For more information, see section 6.2.1.1 – .image.adj on page 80
`..focus`	entry	-	CA	
`...auto`	bool	r	CA	Run an autofocus sequence. Will read true as long autofcus is running.
`...motor`	int32	rwsp	CA	Run focus motor. 0: stop 1 – 100: speed value focusing far in % of full speed -1 – -100: speed value focusing close in % of full speed For A20 with motor focus, the % is N.A, only the sign is of interest, speed control not possible
`...motorPresent`	bool	r	CA	Tells if motor focus function exists
`...pos`	int32	rw	CA	Motor position (0..16000) high = far
`..mfunc`	entry	-		**SEE ALSO:** For more information, see section 6.2.1.3 – .image.mfunc on page 83
`..objpar`	entry	-		
`...ambTemp`	double	rwp		Reflected ambient temperature in K
`...atmTemp`	double	rwp		Atmospheric temperature in K
`...distance`	double	rwp		Object distance (m)
`...emissivity`	double	rwp		Object emissivity; 0.01—1.0

Name	Type	Attr.	com	Description; allowed values
`...extOptTemp`	double	rwp		External optics temperature (K)
`...extOptTrans`	double	rwp		External optics transmission 0.01 - 1.0
`...relHum`	double	rwp		Relative humidity; 0.0—0.99
`..status`	entry	-	M	This Entry contains information about status for certain camera parameters used mainly for the userweb interface.
`...imgtime`	int32	r	M	Time for actual image
`...tmax`	double	r	M	Temperature scale limit
`...tmaxclip`	double	r	M	Temperature max-clip limit used for javascript form validation
`...tmin`	double	r	M	Temperature scale limit
`...tminclip`	double	r	M	See tmaxclip above.
`...tminspan`	double	r	M	Minspan in (dKelvin). Minspan for auto adjust is 2X tminspan
`..zoom`	entry	-	IC	
`...factor`	double	rwp	IC	Zoom factor ▪ 3QWIP 1.0/2.0/4.0 ▪ P/S series 1.0-8.0 ▪ Not implemented in A series

6.2.1.1 .image.adj

Name	Type	Attr.	com	Description; allowed values
`..image.adj`	entry	-		
`...aadjust`	bool	(r)w		Simple adjust; true

Name	Type	Attr.	com	Description; allowed values
`...forceLinear`	bool	rwsp		Force linear color distribution (linearity type is taken from “latchMode” below).
`...freeze`	bool	(r)w		Freeze image; true/false
`...high`	int32	rw		Level + span/2
`...highP`	double	rw		LevelP + spanP/2
`...highT`	double	rw		LevelT + spanT/2
`...latchMode`	ascii	rwp		Internal latch entity; “pixel”, “power”, “temp”. Refers to **Image Settings** in the A-series camera GUI.
`...level`	int32	rw		Image level
`...levelP`	double	rw		Image level in power units
`...levelT`	double	rw		Image level in temperature
`...low`	int32	rw		Level – span/2
`...lowP`	double	rw		LevelP – spanP/2
`...lowT`	double	rw		LevelT – spanT/2
`...span`	int32	rw		Image span
`...spanP`	double	rw		Image span in power units
`...spanT`	double	rw		Image span in temperature
`...auto`	entry	-		
`....cont`	bool	rwsp		Continuous adjust; true/false
`....mode`	ascii	rwsp		Continuous adj mode; ”level”, ”levelSpan”, “histogram”, “levelHist”, “onlyHist”
`....now`	bool	(r)ws		Perform adjust; true

Name	Type	Attr.	com	Description; allowed values
`...palette`	entry	-		
`....aboveBelowColor`	bool	rw		Use above/below coloring; true/false. Defined in palette.
`....description`	ascii	r		Current palette description
`....filename`	ascii	rwp		Current palette filename

*) Resources marked * are not present or not supported in non measure configurations

6.2.1.2 .image.file

Name	Type	Attr.	com	Description; allowed values
`..file`	entry	-		
`...autostore`	bool	rw		Trig resource. When set to true, resource autoname will be trigged, and then store will be used with the generated name. An autonamed image will be stored in currDir
`...currDir`	ascii	rwp		Directory used for autostore and autoname
`...deleteDir`	ascii	rw		If set to a directory name path, the content in that path will be deleted. Use with extreme care (!)
`...imgCount`	int32	rwp		Number used in auto namng
`...naming`	ascii	rwp		Auto name algorithm; “counter”, “directory” or “date”.
`...recall`	ascii	rw		recall file to frame buffer; any filename
`...store`	ascii	rw		store frame buffer to file; any filename

6.2.1.3 .image.mfunc

Name	Type	Attr.	com	Description; allowed values
`..image.mfunc`	entry			
`...cnt`	int32	r		Measure count since start Incremented every time new results are calculated
`...mperiod`	int32	rwsp		Measure interval in 1/100 seconds. Default value 33. (measure loop runs 3 times/second). Avoid values lower than 10.
`...diff`	entry	c		
`....1`	index	dp		1..<configured max num>
`.....active`	bool	rwp		diff value active; true/false
`.....defid0, defid1`	int32	rwp		default ID number for measurement function of 1:st and 2:nd term (i.e “2” for spot 2, mbox 2)
`.....defres0, defres1`	ascii	rwp		default Sub result name for diff terms: “value”, “max”, “maxpos”, “min”, “minpos”, “avg”, “sdev”, “med” (i.e. “min” for “mbox 1 min”)
`.....deftype0, deftype1`	int32	rwp		Default type of measurement function for diff terms: "spot", "mbox", mcircle", "mline", “isotherm”, "reftemp" (i.e. “mbox” for “mbox 2 min”)
`.....id0, id1`	int32	rwp		index of first/second func

Name	**Type**	**Attr.**	**com**	**Description; allowed values**
`.....res0, res1`	ascii	rwp		result type of first/second func “value”, “max”, “maxpos”, “min”, “minpos”, “avg”, “sdev”, “med”
`.....type0, type1`	ascii	rwp		type of first/second func "spot", "mbox", mcircle", "mline", “isotherm”, "reftemp"
`.....valueT`	double	r		read diff temperature (kelvin)
`.....valueValid`	ascii	r		diff results validity symbol, see ...mbox.1.avgValid
`...isotherm`	entry	c		
`....1`	index	dp		1..<configured max num>
`.....active`	bool	rwp		Isotherm active; true/false
`.....attr`	ascii	rwp		"solid", "transparent", "unchanged". solid - Pixel values that match isotherm criterias are given color YCrCb defined by "color" . transparent - Pixel values that match isotherm criterias are given CrCb values from "color". Y varies from 0 to 255 between low and high unchanged - Pixel values that match isotherm criterias are given CrCb values from "color". Y is unchanged.

Name	Type	Attr.	com	Description; allowed values
`.....color`	ascii	rwp		"palette1", "palette2","red","green",blue",yellow","cyan", "magenta","gray"
`.....high`	int32	rwp		Isotherm limit or upper limit
`.....highT`	int32	rw		Isotherm limit/upper limit in termperature
`.....low`	int32	rwp		Isotherm lower limit
`.....lowT`	int32	rw		Isotherm lower limit in temperature
`.....type`	ascii	rwp		“below”, “above”, “interval”, (“dualAbove”, “dualBelow”)
`...mbox`	entry	c		
`....1`	index	dp		1..<configured max num>
`.....active`	bool	rw		Measure box active; true/false
`.....ambTemp`	double	rw		This mbox ambient temperature, if private active (parMask)
`.....avgS`	double	r		Measured average mapped signal level result
`.....avgT`	double	r		Measured average temperature result (kelvin)

Name	Type	Attr.	com	Description; allowed values
`.....avgValid`	ascii	r		average results validity symbol. "=" : result(s) valid "U" : result(s) undefined "<" : result(s) below calibrated limit ">" : result(s) above calibrated limit "*" : result(s) above/below calibrated accuracy limit >,<,* are meaningful for temperature results only. Signal result is always "valid" when not "U".
`.....distance`	double	rw		This mbox object distance, if private active (parMask)
`.....emissivity`	double	rw		This mbox emissivity, if private active (parMask)
`.....height`	int32	rw		size of mbox
`.....isoCoverage`	double	r		Isotherm coverage percentage (0 – 100)
`.....isoValid`	ascii	r	l	isoCoverage validity symbol, see avgValid
`.....maxS`	double	r		Measured maximum mapped signal result
`.....maxT`	double	r		Measured maximum temperature result (kelvin)
`.....maxValid`	ascii	r		max results validity symbol, see avgValid
`.....maxX`	int32	r		X-coordinate of found max signal
`.....maxY`	int32	r		Y - " -

Name	Type	Attr.	com	Description; allowed values
`.....medianS`	double	r		Measured median mapped signal result
`.....medianT`	double	r		Measured median temperature result (kelvin)
`.....medianValid`	ascii	r		median results validity symbol, see avgValid
`.....minS`	double	r		Measured minimum mapped signal result
`.....minT`	double	r		Measured minimum temperature result (kelvin)
`.....minValid`	ascii	r		min results validity symbol, see avgValid
`.....minX`	int32	r		X-coordinate of found min signal
`.....minY`	int32			Y - " -
`.....parMask`	int32	rw		Mask that gives private objpar status. One bit each for emissivity, ambTemp and distance bit 0 set - use local emissivity bit 1 set - use local distance bit 2 set - use local ambTemp For unset bits, emissivity,distance,ambTemp are taken from global values read in .image.objpar

Name	Type	Attr.	com	Description; allowed values
`.....prioRes`	ascii	rw		“max”, “min”, “avg”, ”iso”. To be used by GUI etc. to prioritize what to show as "the result". No other usage. Non measurement configurations only accepts “iso”
`.....sdevS`	double	r		Measured standard deviation mapped signal result
`.....sdevT`	double	r		Measured standard deviation temperature result (kelvin)
`.....sdevValid`	ascii	r		sdev results validity symbol, see avgValid
`.....width`	int32	rw		size of mbox
`.....x`	int32	rw		Upper left corner of mbox
`.....y`	int32	rw		- " -
`...reftemp`	entry	c		
`....1`	index	dp		1..<configured max num>
`.....active`	bool	rwp		ref active; true/false
`.....valueT`	double	r		read reference temperature (kelvin)
`.....valueValid`	ascii	r		ref results validity symbol, see ...mbox.1.avgValid
`...spot`	entry	c		
`....1`	index	dp		1..<configured max num>
`.....active`	bool	rw		Spot active; true/false
`.....ambTemp`	double	rw		This spot ambient temperature, if private active (parMask)
`.....distance`	double	rw		This spot object distance, if private active (parMask)

Name	Type	Attr.	com	Description; allowed values
`.....emissivity`	double	rw		This spot emissivity, if private active (parMask)
`.....parMask`	int32	rw		Mask that gives private objpar status. One bit each for emissivity, ambTemp and distance. See ...mbox.1.parMask above
`.....valueS`	double	r		Measured spot mapped signal result
`.....valueT`	double	r		Measured spot temperature result (kelvin)
`.....valueValid`	ascii	r		spot results validity symbol, see ...mbox.1.avgValid
`.....x`	int32	rw		Position of spot meter X-coordinate
`.....y`	int32	rw		- " - Y-coordinate

6.2.2 .monitor

This part reflects the alarm monitor functionality. Most of this is only implemented in the A-Series, but certain parts applies to E- and P-series

Name	Type	Attr.	com	Description; allowed values
`.monitor`	entry	-	M	
`..alarm`	entry	-	M	
`...1`	entry	-	M	1..<configured max num>
`....a_disableNuc`	bool	rwp	A	Controls if trigged alarm should disable nuc
`....a_markImage`	bool	rwp	A	Controls if trigged alarm should mark (firewire) image
`....a_storeImg`	bool	rwp	A	Controls if trigged alarm should store image
`....active`	bool	rwp	M	Controls if this alarm node is active
`....condition`	ascii	rwp	M	“greater”, ”less”, ”high”, “low”. Condition for alarm function. greater/less applies to “analog” alarm sources. high/low applies to “digital” alarm sources
`....hysteresis`	double	rwp	M	[-10000, 10000] Kelvin for temperature comparations
`....minDuration`	double	rwp	M	[-10000, 10000] seconds
`....result`	bool	r	M	Calculated output. true if alarm condition is satisfied in last calculation
`....resultValid`	ascii	r	M	result validity symbol. See .image.mfunc.X.mbox.avgValid

Name	Type	Attr.	com	Description; allowed values
`....src`	ascii	rwp	M	“spot”, “mbox”, “diff”, “anlgIn”, “intTemp”, “digIn”, “digBiIn”
`....srcId`	int	rwp	M	1..<configured src maxnum>
`....srcRes`	ascii	rwp	M	“value”, “max”, “min”, avg”, “iso”
`....srcUnit`	ascii	rwp	M	“temp”, tempDiff”, “noUnit”, “boolean”, “percentage”
`....threshold`	double	rwp	M	[-10000, 10000] Alarm limit. Kelvin for temperature comparations
`..alarmsCalculated`	bool	rw	M	
`..batchMode`	bool	rwp	M	true if batchMode is active
`...commit`	bool	rw	A	
`...extOptTemp`	ascii	rwp	A	Allowed values: “none”,”anlgIn”, “intTemp” Set if input value should be connected to external optics
`..log`	entry	-	A	
`...clear`	bool	rw	A	Clears log after commit
`...commit`	bool	rw	A	Commits prepared order (clear or file)
`...file`	ascii	rw	A	File name to dump log into
`...fileAttr`	ascii	rw	A	“ovwr”, “append”

Name	Type	Attr.	com	Description; allowed values
`...imageSize`	int32	rwp	A	Total number of images in log. Changes will be applied after system restart. Changed value will then clear all log
`...logSize`	int32	rwp	A	Number of log entrys in log Changes will be applied after system restart. Changed value will then clear all log
`...removeDumped`	bool	rwp	A	Removes filed entries from log
`..output`	entry	-	A	
`...anlg1Id`	int32	rwp	A	[1..10]
`...anlg1Res`	ascii	rwp	A	"value", "max", "min", "avg", "iso" Note, configuration dependent
`...anlg1Src`	ascii	rwp	A	"spot", "mbox", "diff" "anlgIn", "intTemp", “none” Note, configuration dependent
`...anlg2Id`	int32	rwp	A	`See anlg1Id`
`...anlg2Res`	ascii	rwp	A	`See anlg2Res`
`...anlg2Src`	ascii	rwp	A	`See anlg1Src`
`...commit`	bool	rw	A	Apply changes in .monitor.output

Name	Type	Attr.	com	Description; allowed values
`...dig1`	int32	rw	A	[0..6 (..10)] alarm result to put on digital 1 output. 0 means alarm no connection
`...dig2`	int32	rwp	A	[0..6 (..10)] digital 2 output. See dig1
`...dig3`	int32	rwp	A	[0..6 (..10)] digital 3 output. See dig1 Note that .system.ioports.vsync_inactive may override this
`...digbidir`	int32	rwp	A	bidirectional ioport output (when configured as output) See dig 1
`...enabled`	bool	rwp	A	True if alarm/measurement output should be enabled
`..resetAlarms`	bool	rw	M	Reset all alarms when set to true
`..trigger`	entry	-	M	All resources under `..trigger` refer to **Batch Settings** in the camera GUI.
`...1`	entry	-	M	[1..1]
`....condition`	ascii	rwp	M	“greater”, ”less”, ”high”, “low”. Condition for batch condition function. greater/less applies to “analog” alarm sources. high/low applies to “digital” alarm sources

Name	Type	Attr.	com	Description; allowed values
`....hysteresis`	double	rwp	M	[-10000, 10000]
`....result`	bool	r	M	Calculated output. true if batch alarm condition is satisfied in last calculation
`....resultValid`	ascii	r	M	result validity symbol. See .image.mfunc.X.mbox.avgValid
`....src`	ascii	rwp	M	“spot”, “mbox”, “diff”, “anlgIn”, “intTemp”, “digIn”, “digBiIn”
`....srcId`	int	rwp	M	1..<configured src maxnum>
`....srcRes`	ascii	rwp	M	“value”, “max”, “min”, avg”, “iso”
`....srcUnit`	ascii	rwp	M	“temp”, tempDiff”, “noUnit”, “boolean”, “percentage”
`....threshold`	double	rwp	M	[-10000, 10000] Alarm limit. Note, kelvin for temperature comparations

6.2.3 .gui

Name	Type	Attr.	com	Description; allowed values
`.gui`	entry	-		
`..button`	ascii	rw		Usable for button simulation; set to “e” means <yes> button pressed... e, c, s, t, l, r, u, d,
`..local`	entry	-		
`...dateFormat`	ascii	rwsp		Date format: “SI4” (YYYY-MM-DD), “SI2 (YY-MM-DD), ”European” (DD/MM/YY), ”American” (MM/DD/YY)
`...distUnit`	ascii	rwsp		Distance unit; “metric”, ”imperial”
`...language`	ascii	rwsp		Language; ex. “English”
`...languageFile`	ascii	rwp		Language file. Set if external language modules. Ex. “French.FLF”. Needs a GUI restart
`...timeFormat`	ascii	rwsp		Time format; “24hour”, “AM/PM”
`..system`	entry	-		
`...hideGraphics`	bool	rws		true if graphics are hidden. Not used in E series.
`...reset`	bool	(r)ws		Trig resource. Restart the gui.

6.2.4 .config

Name	Type	Attr.	com	Description; allowed values
`.config`	entry	-		
`..calib`	entry	-		
`..image`	entry	-		
`...adjust`	entry	-		
`....histogram`	ascii	r		
`...autofocus`	bool	r		If auto focus function is allowed/supposed to work
`...digital`	entry	r		
`....1394Researcher`	bool	r		Allows FireWire format FM70 (ThermaCAM researcher)
`....dcam`	bool	r		Allows possible non measurement DCAM formats
`....dv`	bool	r		Allows FireWire format DV (if supported by hardware)
`....enable`	bool	r		Master enable for all firewire functionality
`....ip`	bool	r		Enables 1394 ip (FLIR version)
`....maxFrameRate`	int32	r		Allowed highest framerate 0-hw limit only, 1-half, 2-1/4...
`...file`	entry	-		
`....16bitOK`	bool	r		Radiometric storage possible
`....active`	bool	r		File store/recall (from GUI) possible

Name	Type	Attr.	com	Description; allowed values
....directories	bool	r		If sub directories should be possible
....intstorage	bool	r		Internal storage allowed
....maxCount	int32	r		Max number of images allowed to store; 0 = unlimited
....saveInMenu	bool	r		If save should be present as selection in menu
...lockedLens	ascii	r		lens branch name if locked
...maxzoom	int32	r		
...satcolors	bool	r		
...visual	bool	r		
...zoom	ascii	r		“none”
..measure	entry	-		
...diff	entry	-		
....maxnum	int32	r		
...emisstable	bool	r		If emiss table functon (GUI) should be possible
...isoCoverage	bool	r		If isoCoverage result should be possible
...isotherm	entry	-		
....dual	bool	r		
....interval	bool	r		
....maxnum	int32	r		
....multicolor	bool	r		
...maxminmrkOK	bool	r		
...mbox	entry	-		
....maxnum	int32	r		

Name	Type	Attr.	com	Description; allowed values
....resizeable	bool	r		
...mcircle	entry	-		
....maxnum	int32	r		
...mline	entry	-		
....maxnum	int32	r		
....resizeable	bool	r		
...spot	entry	-		
....maxnum	int32	r		
....moveable	bool	r		
...objDist	bool	r		
...objParams	bool	r		
...tAmbient	bool	r		
...tempOK	bool	r		System handles temperature measurement
..monitor	entry	r		
...alarm	entry	r		
....maxnum	int32	r		Max number of alarms
..name	ascii	r		Configuration name
..power	entry	-		
...laser	bool	r		
..revAddOn	ascii	r		
..revision	ascii	r		
..softImageStore	entry	-		
...naming	int32	r		
...opendialogVisible	bool	r		
...GUI	entry	-		

Name	Type	Attr.	com	Description; allowed values
`....format`	bool	r		
`....naming`	bool	r		
`..softMeasure`	entry	-		
`..test`	entry	-		
`...coolerStat`	bool	r	I?	

6.2.5 .system

Name	Type	Attr.	com	Description; allowed values
`.system`	entry	-		
`..autoverify`	int32	rw		
`..autosync`	int32	rwp	A	Controls background storage of current settings (default.reg). 0 – No automatic storage 1 – Storage on change (default) 2 – Set value – Store now, do not change autosync value
`..dsoverride`	ascii	rwp	M	If false, uses detrange as detector setting at restart. If true, override with default detector settings
`..filesUpdated`	int32	rws		This resource should be updated by all application tasks after creating/deleting files in the file system(s), typically image store/delete. The resource could be examined by external software to trig refresh of shown file hierarcy display.
`..restart`	bool	(r)w		Restart system ; true
`..sync`	bool	(r)w		Sync filesystem; true
`..ioports`	entry		A	
`...anlgin`	double	r		Analog in readout value (scaled)
`...anlginHigh`	double	rwp		Scale factor; value for highest analog in voltage

Name	Type	Attr.	com	Description; allowed values
`...anlginLow`	double	rwp		Scale factor, value for lowest analog in voltage
`...anlginS`	int32	r		Analog in raw (A/D) value
`...anlgout1`	double	rw		Analog out 1 write value (scaled)

`...anlgout1High,`
`anlgout1Low`
	double	rwp		Scale factors, analog 1 out write value

`...anlgout1S`	int32	rw		Analog out 1 raw (A/D) write value

`...anlgout2,`
`anlgout2High,`
`anlgout2Low,`
`anlgout2S`
				Analog out 2; See anlgout1<> above

`...digbidir`	bool	rw		
`...digbidir_out`	bool	rwp		
`...digin1`	bool	r		Digital in readout value

`...digout1,`
`digout2,`
`digout3`
	bool	rw		Digital out 1-3 current write value.
				Possible to set if corresponding .monitor.output.digX is 0

`...vsync_inactive`	bool	rwp		false if digital out 3 should output a video v-sync signal
`..rtp`	entry		A	
`...downsample`	int32	rwp		Transfered frame size 1 -> 320x240, 2->160x120
`...enable`	bool	rwp		
`...frequency`	int32	rwp		Images per second

Name	Type	Attr.	com	Description; allowed values
`...mcaddress`	ascii	rwp		Multicast address
`...mcenable`	bool	rwp		Multicast enable
`...port`	int32	rwp		
`...pt`	int32	rwp		Protocol Identifier to use for RTP (Normally 112)
`...ttl`	int32	rwp		
`..statistics`	entry		M	
`...flashWrite`	int32	r		Number of writes to flash memory
`...init`	ascii	r		Date of initiation
`...lcdtime`	int32	r		
`...maxTemp`	int32	r		Maximum sampled temperature for camera (C)
`...minTemp`	int32	r		Minimum sampled temperature
`...powerDown`	int32	r		Number of power-downs
`...runtime`	int32	r		Total runtime (seconds)
`...shutter`	int32	r		Number of total shutter activations
`...startUp`	int32	r		Number of startups
`...uptime`	int32	r		Runtime since last startup (seconds)
`..tempsens`	entry	-		Temperature sensors
`...capsule`	double	rw	M	averaged capsule sensor (kelvin)
`...lens`	double	rw		averaged lens sensor
`...shutter`	double	rw		averaged shutter sensor
`..time`	entry	-		

Name	Type	Attr.	com	Description; allowed values
`...control`	int32	rw		Set to 1 to be able to update sec..year (disconnects sec..year from real time) Set to 2 to insert changed values sec..year as current time. Reads 0 if resources sec..year reflects real time Reads 1 if set to 1 Note, there is no way to return to real time connection after setting to 1 except setting to 2. This means that new time must be entered
`...day`	int32	rw		
`...hour`	int32	rw		
`...min`	int32	rw		
`...month`	int32	rw		
`...sec`	int32	rw		
`...year`	int32	rw		
`..vols`	entry	-		
`...1`	index	dp		1 - internal flash disk, 2 - ram disk, 3 - optional external CFdisk
`....free`	int32	r		Free space in bytes
`....mounted`	bool	r		Mounted; true/false
`....name`	ascii	rwp		Volume name, i.e “28.0/”
`....total`	int32	r		Total bytes in volume
`....used`	int32	r		Used bytes in volume

Name	**Type**	**Attr.**	**com**	**Description; allowed values**
`..web`	entry		M	
`...diff`	ascii			
`...isotherm`	ascii			
`...mbox`	ascii			
`...restabDelay`	int32	rwp		Image viewer. Refresh rate for result table in seconds. 0 means manual update. Then a refresh button is made visible on image viewer page. Do not set to less than 5 (seconds)
`...showAlarm`	bool	rwp		
`...showDiff`	bool	rwp		
`...showIso`	bool	rwp		
`...showMbox`	bool	rwp		
`...showSpot`	bool	rwp		
`...statusDelay`	int32	rwp		

6.2.6 .version

Name	**Type**	**Attr.**	**com**	**Description; allowed values**
`.version`	entry	-		
`..hw`	entry	-		
`...0`	entry	-		
`....article`	ascii	rw		Article
`....name`	ascii	r		
`....revision`	ascii	rw		
`....serial`	ascii	rw		Serial number
`..product`	entry	-		
`...article`	ascii	rw		
`...date`	ascii	rw		
`...name`	ascii	rw		
`...revision`	ascii	rw		
`...serial`	ascii	rw		
`..sw`	entry	c		
`...0`	index	dp		
`....build`	ascii	r		Build number
`....date`	ascii	r		
`....maker`	ascii	r		
`....name`	ascii	r		S/W component name, i.e. “boot2”, “appl”
`....ver`	ascii	r		Version number, typically x.x.x (ex “1.0.6”)

6.2.7 .active

This part mostly contains data extracted from the calibration for the current calibration case.

Name	Type	Attr.	com	Description; allowed values
`.active`	entry	-		
`..switch`	entry	-		Calib case selection
`...ds`	ascii	rw		Detector setting selector. Corresponds to **Range** in camera GUI.
`...fi`	ascii	rw		Filter selector
`...le`	ascii	rw		Lens selector
`...commit`	bool	rw		Switch to selected calibration case

6.2.8 .net

This part reflects network parameters set.

Name	Type	Attr.	com	Description; allowed values
`.net`	entry	-		
`..LANactive`	bool	rw		true if ethernet enabled
`..MACaddr`	ascii	rw		"Undefined" or XX:XX:XX:XX:XX:XX hex number. MAC address to use on ethernet
`..domain`	ascii	rw		domain name reported
`..gateway`	ascii	rw		ip number for gateway XX.XX.XX.XX
`..name`	ascii	rw		name exported on DHCP (1394ip or USB only)
`..interface`	entry			
`...1`	index			1..<highest interface>
`....address`	ascii	rw		XX.XX.XX.XX
`....mask`	ascii	rw		XX.XX.XX.XX

This manual was produced using XML – Extensible Markup Language. For more information about XML, point your browser to: http://www.w3c.org/XML/

Description	Software	Supplier	URL
Version control	ExcoConf	Excosoft	http://www.excosoft.se/eweb/site/exc_pd.html
Editing environment	XML Client	Excosoft	http://www.excosoft.se/eweb/site/excoconf_pd.html
Preformatting	ExcoForm	Excosoft	http://www.excosoft.se/eweb/site/home.html
XML parser	Xerces	Apache	http://xml.apache.org/xerces-j
XSLT processor	Xalan	Apache	http://xml.apache.org/xalan-j
XSL-FO rendering engine	XEP	RenderX	http://www.renderx.com

The following file identities and versions were used in this manual:

(F0001)20248803.xml;4
(F0001)20249303.xml;3
(F0014)20250803.xml;3
(F0014)20250903.xml;5
(F0014)20251103.xml;5
(F0014)20251203.xml;3
(F0014)R0035.rcp;2
(manbase)20236703.xml;18
(manbase)20250403.xml;3

FLIR Systems AB
World Wide Thermography Center
P.O. Box 3
SE-182 11 Danderyd
Sweden
Tel.: +46 (0)8 753 25 00
Fax: +46 (0)8 753 23 64
E-mail: sales@flir.se
Web: www.flir.com

FLIR Systems Inc.
Corporate headquarters
16505 SW 72nd Avenue
Portland, OR. 97224
USA
Tel.: +1 503 684 3731
Fax: +1 503 684 5452
Web: www.flir.com

FLIR Systems Sarl
18 rue Hoche BP 81
F-92134 Issy les Moulineaux
Cedex
France
Tel.: +33 (0)1 41 33 97 97
Fax: +33 (0)1 47 36 18 32
E-mail: info@flir.fr
Web: www.flir.fr

FLIR Systems GmbH
Berner Strasse 81
D-60437 Frankfurt am Main
Germany
Tel.: +49 (0)69 95 00 900
Fax: +49 (0)69 95 00 9040
E-mail: info@flir.de
Web: www.flir.de

FLIR Systems Ltd.
5230 South Service Road, Suite #125
Burlington, ON. L7L 5K2
Canada
Tel: 1-800-613-0507 X30
Fax: 905-639-5488
E-mail: IRCanada@flir.com

FLIR Systems Ltd.
2 Kings Hill Avenue – Kings Hill
West Malling
Kent, ME19 4AQ
United Kingdom
Tel.: +44 (0)1732 220 011
Fax: +44 (0)1732 843 707
E-mail: sales@flir.uk.com
Web: www.flir.com

FLIR Systems S. r. l.
FLIR Systems S.r.l.
Via L. Manara, 2
20051 Limbiate (MI)
Italy
Tel. +39 02 99 45 10 01
Fax +39 02 99 69 24 08
E-mail: info@flir.it
Web: www.flir.it

FLIR Systems Co. Ltd.
Room 1613–15, Tower 2
Grand Central Plaza
138 Shatin Rural Committee Rd
Shatin, N.T.
Hong Kong
Tel.: +852 27 92 89 55
Fax: +852 27 92 89 52
E-mail: flir@flir.com.hk
Web: www.flir.com.hk

FLIR Systems AB
Uitbreidingstraat 60–62
B-2600 Berchem
Belgium
Tel.: +32 (0)3 287 87 11
Fax: +32 (0)3 287 87 29
E-mail: info@flir.be
Web: www.flir.be

FLIR Systems Inc.
USA Thermography Center
16 Esquire Road
North Billerica, MA. 01862
USA
Tel.: +1 978 901 8000
Fax: +1 978 901 8887
E-mail: marketing@flir.com
Web: www.flir.com